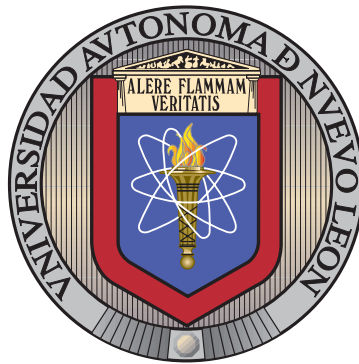


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



PREDICCIÓN DE DEMANDAS ELÉCTRICAS
UTILIZANDO REDES NEURONALES

POR

HÉCTOR HUGO GARCÍA LÓPEZ

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE
MAESTRÍA CIENCIAS DE LA INGENIERIA
CON ORIENTACIÓN EN SISTEMAS

JULIO 2021

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



PREDICCIÓN DE DEMANDAS ELÉCTRICAS
UTILIZANDO REDES NEURONALES

POR

HÉCTOR HUGO GARCÍA LÓPEZ

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE
MAESTRÍA CIENCIAS DE LA INGENIERIA
CON ORIENTACIÓN EN SISTEMAS

JULIO 2021



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN





FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA


Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica
Subdirección de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis "Predicción de demandas eléctricas utilizando redes neuronales", realizada por el alumno Héctor Hugo García López, con número de matrícula 1484166, sea aceptada para su defensa como requisito para obtener el grado de Maestría en Ciencias de la Ingeniería con orientación en sistemas.

El Comité de Tesis


Dr. Vincent André Lionel Boyer
Director


Dra. María Angélica Salazar Aguilar
Revisor


M. en A. Julio César López González
Revisor

Vo. Bo.


Dr. Simón Martínez Martínez
Subdirector de Estudios de Posgrado



175

San Nicolás de los Garza, Nuevo León, julio de 2021



ÍNDICE GENERAL

1. Introducción	1
1.1. Definición del problema	3
1.2. Objetivo	3
1.3. Hipótesis	4
1.4. Justificación	4
1.5. Metodología	5
1.6. Estructura de la tesis	5
 2. Revisión de literatura	 6
2.1. Series de tiempo	7
2.2. Métodos estadísticos autorregresivos	7
2.3. Redes neuronales artificiales	9
2.4. Tipos de neuronas	11
2.5. Limpieza de datos	12
2.5.1. Univariados	13
2.5.2. Multivariados	13

3. Metodología	15
3.1. Caso de estudio	15
3.2. Análisis	16
3.2.1. Gráficas de área	18
3.2.2. Análisis de correlación	19
3.2.3. Prueba de Dickey-Fuller	21
3.3. Gráfica por intervalo de tiempo	23
3.4. Limpieza de datos	23
3.4.1. Rango intercuartil	26
3.4.2. Local Outlier Factor	27
3.4.3. PCA	27
3.5. Análisis posterior a la limpieza de datos	30
3.6. Descomposición de series de tiempo	30
4. Experimentos y resultados	34
4.1. Preparación de los datos	34
4.2. Modelos preliminares	36
4.3. Modelos finales	44
5. Conclusiones y recomendaciones	54
5.1. Recomendaciones	54
5.2. Trabajo a futuro	55

A. Apéndice	56
A.1. Dickey-Fuller	56
A.2. Descomposición de serie de tiempo	56
A.3. Número de capas en las redes neuronales	57
A.4. Tipo de escala	58
A.5. Hiperparámetros en las redes neuronales artificiales	59
A.5.1. Epochs	59
A.5.2. Dropout	59
A.5.3. Tamaño del batch	60
A.6. Funciones de pérdida	60
A.6.1. Error porcentual absoluto medio	60
A.6.2. Error absoluto medio	61
A.6.3. Error cuadrado medio	61
A.6.4. Criterio de R^2	62

ÍNDICE DE TABLAS

4.1. Ejemplo de los datos utilizados en los modelos.	35
--	----

CAPÍTULO 1

INTRODUCCIÓN

“Nunca se puede predecir un acontecimiento físico con una precisión absoluta.”

MAX PLANCK

El ser humano tiene la necesidad de saber ciertos acontecimientos con antelación, como el clima que habrá ciertos días feriados para saber si irán a la playa o verán películas con chocolate caliente. Saber ciertos acontecimientos con un grado controlado de error cobra una papel importante en las organizaciones ya que puede costar millones de pesos.

Predecir la demanda de un producto para alguna temporada permite a la empresa productora enfocar sus costos, inventarios y recursos humanos en satisfacer dicha demanda de la mejor manera posible. Un ejemplo puede ser el crecimiento de clientes en un restaurante durante el súper bowl, no conocer o tener cierta estimación del crecimiento de clientes puede provocar que no haya suficientes cervezas frías para todos los comensales causando que las cervezas calientes sean metidas a los congeladores o en baños con hielo, agua y sal para bajar la temperatura de esos productos y satisfacer la demanda lo más rápido posible. Los riesgos de una predicción errónea bajo este contexto es tener a un comensal insatisfecho durante algunos minutos, incurriendo en el costo de un producto gratis con la finalidad de complacer

a un cliente que probablemente haya perdido su lealtad. Sin embargo, existen riesgos más altos por no predecir bien una demanda, por ejemplo, en la producción de fármacos o la distribución de agua donde se pone en juego las vidas de las personas.

Una buena predicción bajo estimaciones controladas que permitan mediante la visualización de distintos escenarios tomar decisiones, genera una planeación efectiva de los recursos de una organización, impactando de forma positiva en la logística de la misma. Entendiendo a la logística como el conjunto de conocimientos, acciones y medios destinados a proveer y prever los recursos necesarios que posibiliten realizar una actividad principal en tiempo, forma y al costo más oportuno en un marco de productividad y calidad. La logística es un proceso de la cadena de suministro; Handfield y Nichols Jr (1999) mencionan que la administración de la cadena de suministro abarca todas las actividades relacionadas con el flujo y transformación de bienes, desde la etapa de materia prima hasta el usuario final, así como los flujos de información relacionados.

Existen distintos tipos de cadenas de suministros, dependiendo del giro de la empresa. El sistema de suministro eléctrico es el encargado de estudiar todos los elementos existentes entre la generación de la energía hasta el consumidor final.

Una vez que la energía es generada, se tienen las actividades de transmisión y distribución de energía eléctrica. Para la primer actividad se utilizan subestaciones elevadoras de energía, esto es porque a mayor tensión se pierde menos energía por efecto de la impedancia y el fin de esta actividad es la distribución de energía a zonas cercanas a donde va a ser utilizada por los usuarios. Después que ha llegado a las subestaciones reductoras, que es donde se reduce la tensión, la distribución de energía se realiza a través de las Redes Generales de Distribución (RGD) la cual, con la infraestructura disponible en 2017, se daba servicio a 42.2 millones de usuarios.

Las RGD están compuestas por circuitos que a su vez están conformados por transformadores, líneas, cuchillas, fusibles, switches, restauradores que cada uno opera con distintos niveles de carga, y a diario se hacen modificaciones en la manera en que se conectan éstos componentes con fines de mantenimiento o expansión del

circuito. La decisión de configurar la RGD, es tomada por personal que tiene experiencia, con ayuda de herramientas que permite visualizar la manera en que se conecta la RGD.

Tomar la decisión de qué componente se tiene que configurar para dar un mejor servicio, teniendo en cuenta las reglas de seguridad, es una tarea complicada por lo extenso de sus datos y la variabilidad del consumo. Para tener un ejemplo de lo antes mencionado, se toma como referencia el área metropolitana de Monterrey, dado que Monterrey es una ciudad industrial, hay altos consumos de energía en zonas industriales a lo largo del día, mientras que el consumo fuerte en las zonas residenciales se presenta en las horas nocturnas. Diariamente la empresa encargada de suministrar la energía en Nuevo León se ve en la necesidad de hacer adecuaciones en la red eléctrica por distintos motivos, ya sea por mantenimientos, fallas en los componentes de la red o porque se requiere seguir incrementando las líneas, por lo tanto se tiene que realizar ciertas maniobras para cambiar la forma en que la red está conectada.

1.1 DEFINICIÓN DEL PROBLEMA

Al momento de reconfigurar la RGD, los tomadores de decisiones no cuentan con herramientas cuantitativas para realizar predicciones acerca de la demanda de energía eléctrica, lo que aumenta la dificultad de encontrar una buena configuración. Adicionalmente, existen dificultades en la capacitación del nuevo personal ya que los expertos conocen el comportamiento de las demandas de manera empírica, por la experiencia de desempeñar esa labor por años.

1.2 OBJETIVO

Elaborar una herramienta que permita predecir la demanda de energía eléctrica en un intervalo de tiempo suficientemente amplio y con errores cuantificables.

1.3 HIPÓTESIS

Mediante una arquitectura de red neuronal se puede predecir la demanda de energía eléctrica en alimentadores de la zona metropolitana de Monterrey.

1.4 JUSTIFICACIÓN

Según cifras del Programa de Desarrollo del Sistema Eléctrico Nacional (PRO-DESEN) la longitud de las líneas de distribución ascienden a más de 800 mil kilómetros en toda la república Mexicana, convirtiéndola en un sistema complejo para manejarla. En la RGD y por motivos de seguridad, se tienen ciertos componentes que ayudan a proteger la red y a cambiar su topología, es decir, la forma en que están conectados los componentes. Una peculiaridad de la red es que se puede abastecer de energía eléctrica por diferentes puntos, esto se hace porque si algún componente llega a presentar una falla, no se tiene que privar de electricidad a los usuarios.

El ciclo completo para realizar una reconfiguración en la RGD empieza cuando se tiene la necesidad de hacer una modificación en un circuito, llega el requerimiento al departamento encargado de analizar la topología de la RGD y ellos analizan los posibles escenarios de como se puede configurar la RGD para aislar el componente a cambiar. El tiempo aproximado para realizar el análisis de la RGD y obtener una solución puede tardar semanas, ya que se tiene que considerar diversos factores como ¿Cuál será la carga del (los) circuitos involucrados después de la reconfiguración?, ¿Qué componentes se tienen que modificar para obtener la nueva configuración?, ¿Cuál es una manera eficiente de balancear las cargas entre los alimentadores de manera que se protejan los componentes del circuito?. Para responder éstas preguntas se necesita estimar la demanda que habrá en los alimentadores durante los días en que se realizará la maniobra de configuración, por lo que es necesario una herramienta que permita tener esa información.

1.5 METODOLOGÍA

Para llevar a cabo el proyecto de tesis, se siguió la siguiente metodología: revisamos la literatura para observar qué es lo que se ha hecho para predecir demandas, cuáles son los modelos que permiten realizar predicciones y cuantificar el error, ver las particularidades de los datos y entender las distintas metodologías. Posteriormente, se obtienen los datos que están distribuidos en diferentes bases de datos. Se hace un análisis exploratorio de datos para visualizar la serie de tiempo asociada con las demandas y posteriormente se lleva a cabo la limpieza de los datos.

Una vez que los datos están preparados para la predicción, se hace un diseño de experimentos para ver qué factores ofrecen mejor desempeño en la predicción de las demandas. Después de haber obtenido una arquitectura de red neuronal que satisfaga las restricciones de dicha predicción, se implementa la predicción con datos actuales y se evalúa el desempeño con datos recientes, actualizados.

1.6 ESTRUCTURA DE LA TESIS

El presente documento está compuesto por cinco capítulos. En el capítulo dos se presenta la revisión de literatura, donde se define qué es una serie de tiempo, el trabajo relacionado a las predicciones de series de tiempo y algunos elementos que se encuentran en los diferentes modelos de redes neuronales. La metodología se aborda en el capítulo tres, se proveen detalles de las fuentes de recolección de los datos, el acomodo que se hace y algunos análisis y resultados preliminares. En el capítulo cuarto se muestran los resultados obtenidos con diferentes modelos de redes neuronales sobre el conjunto de datos disponible. Por último, en el capítulo cinco se muestran las conclusiones de esta tesis, así como el trabajo a futuro.

CAPÍTULO 2

REVISIÓN DE LITERATURA

La predicción de demanda energética es un tema ampliamente estudiado en la literatura porque representa una herramienta útil en las empresas así como por la información que aporta acerca del consumo de electricidad en las zonas estudiadas. Existen diferentes algoritmos que se utilizan para predecir las demandas eléctricas, van desde los que utilizan series de tiempo como en Reikard (2009), Despotovic *et al.* (2016), distintos modelos de machine learning tradicionales como en Chia *et al.* (2015), Deo *et al.* (2016) donde se usan máquinas de soporte vectorial o en Voyant *et al.* (2017) y Benali *et al.* (2019) en los cuales el enfoque es a través de árboles de decisión hasta modelos más complejos donde se usan diferentes algoritmos juntos como en Halabi *et al.* (2018), donde el autor combina un modelo ANFIS con optimización por enjambre de partículas, optimización por evolución diferencial y algoritmos genéticos.

Partiendo de estos enfoques ha sido posible estudiar las demandas eléctricas y obtener ciertas pistas o guías para facilitar la construcción de modelos que permitan realizar predicciones de datos futuros con éxito. En esta sección, se analizan los casos de estudio que se relacionan con dicha problemática y algunos resultados obtenidos que permiten tener una visión general de la metodología que se debe tener al momento de abordar los modelos del consumo de energía eléctrica.

2.1 SERIES DE TIEMPO

Las demandas de electricidad son observaciones del consumo de energía eléctrica en un determinado periodo de tiempo. En el municipio de Monterrey, en el estado de Nuevo León, se sabe que el consumo de electricidad es más alto en los meses donde hace calor con respecto a los meses donde la temperatura es más baja.

Un conjunto de observaciones medidas secuencialmente a través del tiempo es llamado serie de tiempo Chatfield (2000). Las series de tiempo han sido estudiadas por diferentes autores con distintos motivos, tales como, detectar anomalías en videovigilancia Teng (2010), en el contexto de sistemas financieros Kim (2003) o en el diagnóstico de incidentes de servicios en línea Luo *et al.* (2014).

2.2 MÉTODOS ESTADÍSTICOS AUTORREGRESIVOS

Los modelos autorregresivos integrados de medias móviles (ARIMA, por sus siglas en inglés) se han utilizado con objetivos tales como pronosticar cuando un equipo va a dejar de funcionar Ho y Xie (1998), predecir los precios de las acciones en la ciudad de Nueva York Ariyo *et al.* (2014), predecir la producción de energía eléctrica de la ciudad de Nysted en Dinamarca Chen *et al.* (2009), esta variante del método ARIMA (LARIMA) es capaz de predecir mejor la producción de electricidad en comparación con un modelo Markoviano discreto.

Otro ejemplo se presenta en Calheiros *et al.* (2014) donde se utiliza, modelos ARIMA con el fin de mejorar la calidad de servicios de software en la nube al utilizar un modelo que pueda predecir cuando una máquina virtual va a tardar demasiado tiempo en procesar la información de cierta aplicación. Una aplicación que permite predecir el flujo de tráfico que habrá por hora y por día en una red GSM en China con el fin de adaptar las bandas para las horas con alto tráfico es propuesto en Shu *et al.* (2005) donde hacen uso de dos tipos de modelos ARIMA.

Para mejorar la predicción se han implementado otras técnicas que aportan información al modelo ARIMA como en Lee y Ko (2011) donde además del modelo se utiliza un esquema de elevación con el fin de descomponer la serie de tiempo de la demanda y aplicando diferentes modelos ARIMA ajustados a esas sub-series, o en Ong *et al.* (2005) donde se utiliza un modelo ARIMA para predecir el precio de artículos en el sector de semiconductores industriales, aunado a esto se hace uso de modelos genéticos para encontrar el modelo que mejor se ajusta a los datos.

En el ámbito energético se tienen numerosos ejemplos como en Mbamalu y El-Hawary (1993) donde se propone una metodología distinta, la *Box Jenkins*, con el objetivo de encontrar un modelo ARIMA para predecir el consumo de electricidad de una empresa en Nueva Escocia. Otro ejemplo de una empresa en Taiwan se encuentra en Cho *et al.* (1995) en donde se propone un modelo ARIMA y un modelo de función de transferencia para predecir la demanda de una semana en Taipower. También se han hecho mezclas de diferentes modelos con ARIMA, por ejemplo con suavización exponencial, tal es el caso de Taylor (2003) en el que predice la demanda que habrá desde treinta minutos después hasta un día, los datos que se usan son las demandas de energía de Inglaterra y Gales medidas cada 30 minutos. Los modelos ARIMA son utilizados debido a que modelan las series de tiempo mediante técnicas estadísticas en las que, para encontrar los parámetros de la fórmula que modela la demanda se suelen seguir distintas metodologías.

Por lo general, se usa la metodología Box Jenkins Box *et al.* (1967) para determinar los parámetros (p) que representa al modelo autorregresivo (AR) y (q) del modelo de medias móviles (MA). Esta metodología consiste en tres fases, la primera es la de analizar la serie de tiempo buscando si cumple el supuesto de que los datos sean estacionarios, observando los datos y su autorrecursividad para identificar el modelo a emplear AR, MA, ARMA, ARIMA, después hay que investigar los parámetros que se puedan ajustar a los datos y por último, probar el modelo con los datos buscando que los residuos (errores) sigan una distribución normal (media 0, varianza σ^2), éstas fases son iterativas por lo que al no cumplir que los errores distribuyan normal se vuelve al paso uno.

En la literatura se encuentran ejemplos de otras metodologías en las cuales se busca mejorar la calidad del modelo, un ejemplo de esto se da en Wang *et al.* (2012) donde se estudian las demandas del norte de China usando el método de Box-Jenkins y para reducir el error, introducen tres maneras distintas de agregar parámetros al modelo que dependen del error en las predicciones, una optimización por enjambre de partículas aplicadas a series de Fourier, un modelo S-ARIMA para modelar los residuos y una combinación de los residuos de ambos, obteniendo en los tres modelos mejores predicciones que el modelo S-ARIMA implementado.

2.3 REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales son modelos computacionales inspirados biológicamente que consisten en elementos de procesamiento llamados neuronas y conexiones entre ellos con coeficientes ligados a las conexiones Shanmuganathan (2016). Este tipo de modelos se utilizan principalmente para predecir o clasificar.

En el artículo Xu *et al.* (2020) se utilizan las redes neuronales junto con un sistema de lógica difusa para clasificar fallas en equipos, al analizar las vibraciones de los equipos, y se establece que si se logra reducir la tasa de apagado planificado y la del apagado forzado por fallas en un generador de turbina de vapor de 400Mw de capacidad en 1 % se puede ahorrar \$17 millones al año.

En Pang *et al.* (2020) predicen el precio de las acciones por medio de una red neuronal con capas LSTM, se inspiraron en el vector de palabras usado en modelos de aprendizaje profundo para procesar texto, donde cada palabra tiene un vector de ceros y unos, la desventaja de esta representación es que cada elemento tiene una dimensión igual al diccionario que se esté utilizando, aparte que no se ve la relación existente entre las palabras.

En Hannun *et al.* (2019) se utilizan electrocardiogramas para clasificar 12 tipos diferentes de ritmos cardíacos usando aprendizaje profundo, la razón de usar aprendizaje profundo en lugar de una red neuronal que tenga arquitectura más simple es

porque los datos de entrenamiento consisten en 91,232 registros lo que proporciona una gran cantidad de datos para ajustar los parámetros existentes en las redes neuronales de aprendizaje profundo.

En Maiti *et al.* (2020) construyen una arquitectura de red neuronal que tiene la capacidad de predecir el índice de Sharp que es la relación entre la recompensa y la variabilidad de una inversión; el modelo generado puede entender la posición en la que se encuentra el inversor y le genera una prima de riesgo adicional con inversiones del mismo nivel de riesgo. La arquitectura desarrollada tiene una capa de entrada de ocho neuronas, dos capas ocultas de diez neuronas cada una cuyas funciones de activación son sigmoide y una capa de salida. Se construyeron dos redes neuronales para dar paso a una red neuronal de ensamble, cada una de las redes neuronales predice una serie de tiempo y al final se evalúa con un método llamado Delphi donde se tiene en cuenta las predicciones de las redes para decidir cual será la predicción final.

Para predecir cargas a nivel residencial en Kong *et al.* (2017) hacen uso de una red LSTM, estas predicciones son más complejas en el sentido en que al predecir una subestación le suman las cargas individuales de los usuarios, lo que la vuelve más estable en comparación a las cargas por usuario que dependen del comportamiento humano, se utilizaron los datos de cargas de Australia considerando 69 usuarios que fueron elegidos bajo el criterio de que todos tenían un sistema de agua caliente. En la fase de limpieza de datos usaron el algoritmo DBSCAN donde cada punto en el espacio de 48 dimensiones representa un día de carga eléctrica medidas en intervalos de media hora, el optimizador ADAM da mejores resultados que otros optimizadores como SGD, Adagrad, Adadelta y RMSProp.

Utilizando los datos provenientes de 370 subestaciones en Portugal, desde inicios del 2011 hasta el final del 2014, registrados en el repositorio de la Universidad de California en Ertugrul (2016) se hace una validación cruzada de Monte Carlo para la arquitectura de red neuronal con una capa oculta, se comparan dos métodos de entrenamiento: extreme learning machine y recurrent extreme learning machine.

En Houimli *et al.* (2020) se crea una red neuronal cuyo algoritmo de optimización es el de Levenberg-Marquardt, los datos fueron proveídos por una compañía de Túnez que ofrece servicios eléctricos y de gas. Los datos de entrada se clasifican en tres categorías, primero están las cargas del día anterior, luego las condiciones climáticas tales como la temperatura mínima y máxima por día, se realiza un análisis donde se obtiene información acerca del tipo del día, tipo de la semana, tipo del mes y tipo del año. Lo que se quiere predecir, es la carga de media hora durante un cierto tiempo en el día.

Existen muchas combinaciones de factores que pueden cambiar de una red neuronal a otra, desde la función de activación de las neuronas hasta la función que calcula los errores y actualiza los pesos de los parámetros. Al definir una red neuronal también estamos definiendo parámetros que deben ser entrenados, estos parámetros pueden variar según el tipo de neurona que se escoja, las conexiones entre las neuronas y el número de capas. Así mismo, éstos parámetros guardan relación con la cantidad de datos de entrada disponibles, ya que estos datos se utilizarán para actualizar dichos parámetros. A esta búsqueda de los mejores parámetros se le llama hiperparametrización.

2.4 TIPOS DE NEURONAS

Una neurona artificial es la unidad de la que están compuestas las redes neuronales artificiales. Una neurona es un modelo que contiene distintos parámetros y dependiendo del tipo de neurona es el modelo que se encuentra en ella y por ende la cantidad de parámetros que deben de ser ajustados durante el entrenamiento de los modelos.

El funcionamiento general de una neurona artificial es bastante sencillo; para ciertos datos de entrada los cuales son pasados por una función de activación se obtiene un valor de salida. La función de activación se escoge dependiendo del objetivo al momento del entrenamiento de la red neuronal.

Las neuronas artificiales se pueden clasificar con respecto a sus valores de salida. Pueden ser reales o binarias; también llamadas lineales o no lineales. Las salidas de las neuronas binarias solo toman dos valores, puede ser -1, 1 o 0, 1. En el caso de las neuronas reales pueden tomar valores dentro de un intervalo, lo más común es que esos intervalos sean $[-1, 1]$ o $[0, 1]$ aunque también las hay en cualquier otro rango dentro de los reales (Acevedo *et al.*, 2017).

Existen otros tipos de neuronas que no solo son lineales o no lineales y tienen que ver con las conexiones que se permiten en las neuronas. La forma clásica es que las neuronas se conecten con la capa posterior, pero esto se ve limitado cuando se intentan aprender patrones del tiempo (Pose, 2009). A las redes neuronales cuyas neuronas permiten bucles se les llama redes neuronales recurrentes.

Existen diferentes arquitecturas de redes neuronales recurrentes como en (Katte, 2018) donde presenta hasta 16 tipos diferentes. La forma más básica es la RNN (red neuronal recurrente). Las neuronas LSTM (long short term memory) o neuronas de corto y largo plazo son un tipo de neuronas utilizadas en redes neuronales recurrentes, su nombre proviene de que tiene distintos parámetros que permiten a la neurona utilizar datos del pasado en datos a corto o largo plazo. Existen diferentes experimentos que confirman que las redes neuronales LSTM presentan superioridad sobre otros métodos al predecir las demandas eléctricas como en (Abbasimehr *et al.*, 2020).

2.5 LIMPIEZA DE DATOS

El proceso de limpieza de datos es importante porque los datos usualmente están sucios (Dasu y Johnson, 2003) esto puede ser por que provienen de distintos archivos, por diferencias en las mediciones, por algún problema cuando se estaban capturando los datos etc. Debido a que los datos se van a ingresar a un modelo de redes neuronales es recomendable que se ingresen lo más limpios posibles con el fin de que el modelo entienda de una mejor manera las relaciones existentes en los

datos. Por lo que, además de usar las visualizaciones de los datos se hacen ciertos análisis para obtener los registros que presentan cierta incongruencia.

Para poder limpiar los datos con éxito hay que definir la manera en que se encontrarán los datos que se consideran sucios. Se usan distintos algoritmos para encontrar estos datos sucios o comunmente llamados datos atípicos, los algoritmos para encontrarlos son diversos y desde distintos alcances. Por ejemplo, se pueden encontrar algoritmos que detectan datos atípicos en una sola variable, considerando los cuartiles o percentiles así como si los datos provienen de una serie de tiempo entonces se puede trabajar con los percentiles pero por ventanas de tiempo. También existen algoritmos para datos multivariados los cuales consideran vecindades para interpretar si el dato tiene un comportamiento similar al resto o se aleja del comportamiento habitual por lo cual sería un dato atípico. En esta sección se habla de los algoritmos utilizados para detectar los datos atípicos, sus propiedades y su utilidad en los datos con los que se cuentan.

2.5.1 UNIVARIADOS

El boxplot es de los métodos más usuales para visualizar datos univariados y identificar datos atípicos (Verardi y Vermandele, 2018). El algoritmo por rango intercuartil permite un mejor ajuste los datos ya que en este algoritmo no requiere que los datos provengan de una distribución normal. Lo que hace es obtener cuartiles y se le llama rango intercuartílico a la resta del cuartil tres menos el cuartil uno. Este análisis se suele hacer sobre variables aleatorias que no siguen una distribución normal.

2.5.2 MULTIVARIADOS

El algoritmo del factor de valor atípico local propuesto en el 2000 por Markus Breunig (Breunig *et al.*, 2000) está basado en el concepto de densidad local. Supon-

gamos que se tiene un punto en el espacio, se observa la distancia que tiene con respecto a sus tres vecinos más cercanos luego a cada uno de sus vecinos se observa la distancia de los tres vecinos más cercanos a cada uno de ellos; usando estas distancias se puede estimar la densidad de dicho punto. Comparando la densidad local de un punto con respecto a la densidad local de sus vecinos se pueden encontrar los puntos con densidad muy baja los cuales son los datos atípicos.

De las ventajas que se pueden encontrar en este algoritmo es que se basa en las propiedades del espacio en el que se encuentre, al utilizar un concepto de densidad se puede detectar datos alejados de los demás sin necesidad de decidir si pertenece a un subconjuntos de datos o pertenece a otro. Parte de las desventajas que tiene es que la densidad no dice si un punto es atípico o no.

El análisis de componentes principales o PCA, por sus siglas en inglés. El enfoque realizado por este análisis es el de reducir las dimensiones de los datos originales, es decir, representar el conjunto de datos originales creando nuevas variables no correlacionadas. Esta técnica es muy utilizada en los modelos de aprendizaje máquina como fase previa, antes de ingresar los datos a los modelos, se utiliza para poder reducir significativamente la dimensión de los datos comparado con la poca pérdida de información.

CAPÍTULO 3

METODOLOGÍA

Una vez que se conocen los beneficios y bondades que proporcionan los modelos de redes neuronales y teniendo claras las métricas más utilizadas al predecir cuantitativamente la demanda energética, se procede a trabajar con los datos del caso de estudio. Con el objetivo de utilizar una red neuronal como herramienta para predecir la demanda eléctrica se debe analizar información histórica con el fin de identificar patrones y tendencias.

Ingresar solo los datos de la demanda eléctrica diaria no ofrece la información suficiente para que una arquitectura de red neuronal sea capaz de entender el patrón existente en los datos, esto es porque la demanda de energía está relacionada con las costumbres de los habitantes que se están abasteciendo.

En este capítulo se describe el tratamiento que se le da a los datos, se presenta el análisis de los mismos, así como se concluye con los datos que se van a ingresar a los modelos de redes neuronales.

3.1 CASO DE ESTUDIO

El caso de estudio es la empresa encargada de suministrar energía eléctrica en Nuevo León, la Comisión Federal de Electricidad (CFE), la cual cuenta con 103 subestaciones y 182 transformadores, dando un total de 744 circuitos a donde están

conectados los usuarios. Cada subestación tiene un alimentador principal que provee energía a todo el circuito. Se analizan los datos de la subestación llamada “Canadá”, que está en el municipio de Monterrey. La base de datos contiene las demandas de energía eléctrica, medidas cada 10 minutos, en el periodo que comprende de Mayo del 2018 a Junio del 2019.

Una subestación está formada por varios tipos de alimentadores que se pueden clasificar en “alimentadores padre” y “alimentadores hijo”. A los alimentadores padre, les llega la energía en grandes cantidades de voltaje, para transportar la electricidad con el menor desperdicio posible, al llegar a estos alimentadores se baja el voltaje para ser distribuido hacia los alimentadores hijo los cuales abastecen de electricidad a los usuarios finales.

Un alimentador padre abastece entre tres y cinco alimentadores hijo y cada uno de esos alimentadores tiene la capacidad de mandar los datos de la electricidad que está pasando a través de ellos. Como cada alimentador es independiente, algunos envían los datos cada diez minutos, otros cada quince minutos. En la base de datos se guarda el dato de la fecha y hora, la carga eléctrica en ese instante de tiempo y el nombre del alimentador.

3.2 ANÁLISIS

Primero observamos, la cantidad de datos faltantes que existen por mediciones de cada alimentador. Como se puede apreciar en las Figuras 3.1 y 3.2 donde el total de los datos organizados por mediciones de cada 10 minutos en el intervalo de tiempo seleccionado debería ser de 61889 mediciones, teniendo el alimentador con mayor cantidad de datos nulos el alimentador ECN4235 con 583 datos nulos que representa 0.94 % de los registros.

Con el fin de visualizar estos datos, se hacen unas gráficas de línea ver Figura 3.3. Como se puede apreciar cada una cuenta con su propia demanda y su propia escala, por ejemplo, “ECN4155” solo llega hasta 80 kw/h mientras que “ECN4145”

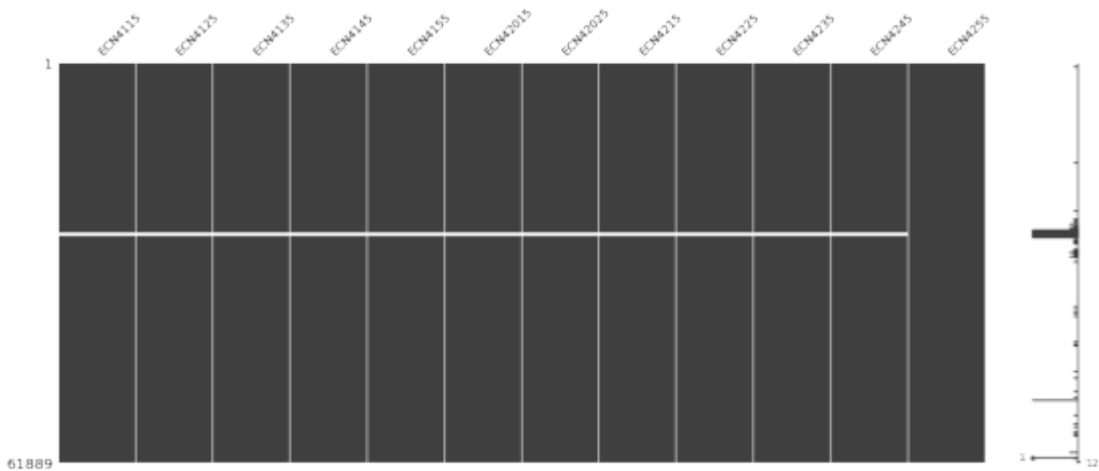


Figura 3.1: Gráficas de posición de datos nulos de alimentadores

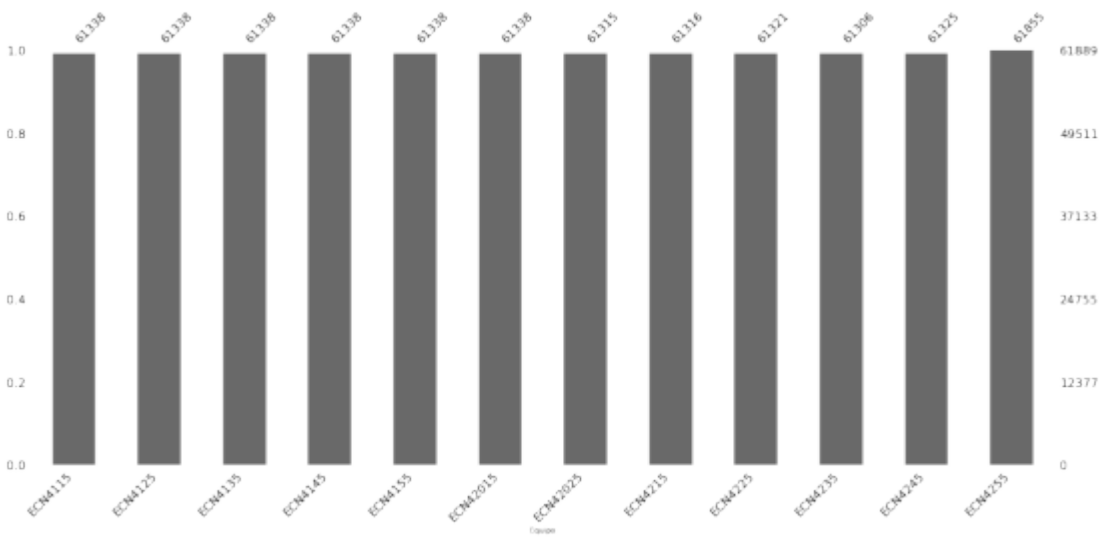


Figura 3.2: Gráficas de cantidad de registros de alimentadores.

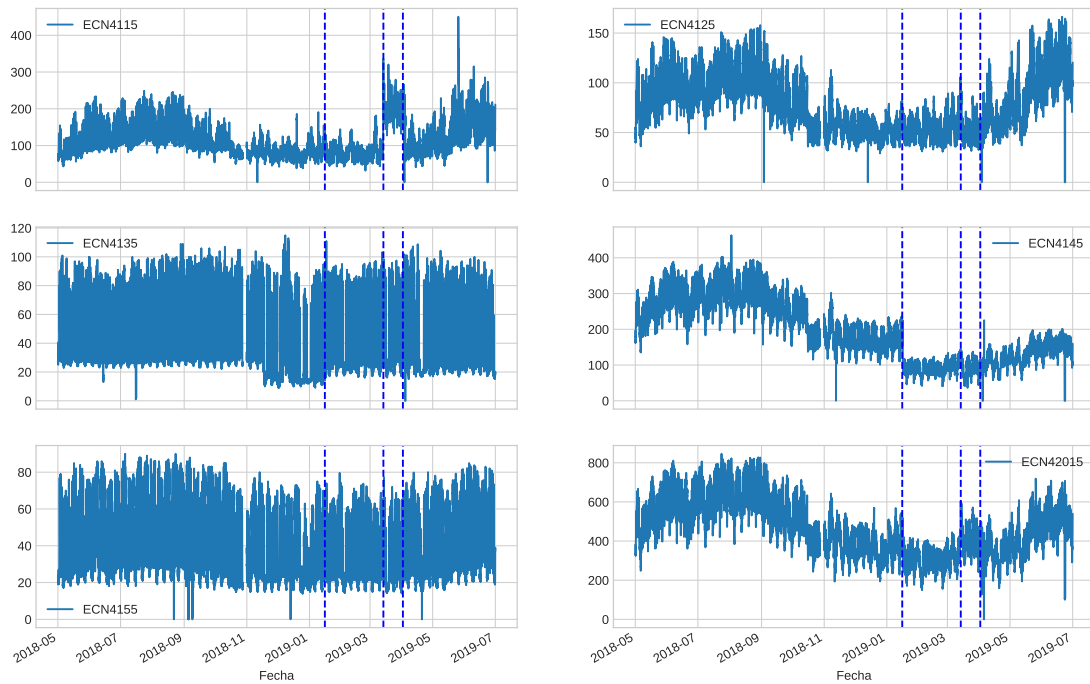


Figura 3.3: Gráficas de alimentadores a través del tiempo.

alcanza los 400 kw/h. Se sabe que el alimentador “ECN45015” es el alimentador padre de los cinco alimentadores hijo que se muestran en la Figura 3.3.

3.2.1 GRÁFICAS DE ÁREA

Las gráficas de área se usan para representar los totales acumulados a lo largo del tiempo, ver Figura 3.4. En la parte de abajo se grafica cada una de las series de tiempo, una encima de la otra. Se aprecia que el acumulado total es de 1600 kw/h y se presenta al rededor de Junio y Julio que son los meses de mayor calor en la ciudad de Monterrey, mientras que el mínimo acumulado es de 800 kw/h y se presenta en los meses de Enero y Febrero que es donde el clima en Monterrey es más frío. En la misma figura pero en la parte de arriba, se tienen las gráficas de área vistas en porcentajes. El cien por ciento está representado por el alimentador padre “ECN45015” que es el que alimenta a los demás alimentadores hijos; se puede ver que desde mayo del 2018 hasta enero del 2019 los porcentajes permanecieron sin

cambios, es decir, el alimentador “ECN4115” representó un veinticinco por ciento del total, mientras que el alimentador “ECN4145” representó un cincuenta por ciento del alimentador padre. Cuando la RGD cambia la forma en que se conecta, esto cambia los porcentajes de cada alimentador, algunos usuarios son cambiados de un alimentador a otro y es por eso que el alimentador “ECN4145” representado en color rojo presentó una disminución, en su porcentaje a partir de enero del 2019 al mismo tiempo que “ECN4125” presentó mayor carga en esas mismas fechas.

Viendo estos comportamientos de las demandas de los alimentadores se decide trabajar con la serie de tiempo del alimentador padre. Un motivo es para acotar este trabajo y el segundo es porque las demandas de los alimentadores hijo representan un porcentaje del alimentador padre, así que teniendo una predicción de este se puede tener información acerca de la demanda de los alimentadores hijo y esto puede ayudar a tomar la decisión de cómo reconfigurar la RGD.

3.2.2 ANÁLISIS DE CORRELACIÓN

La corriente en cada alimentador es “trifásica” lo que quiere decir que está formado por tres corrientes alternas monofásicas, por lo que en realidad se tienen tres demandas, una para cada fase. Se hace un análisis de correlación de Pearson que es una medida de dependencia lineal entre distintas variables aleatorias, estas son cada una de las tres fases, el coeficiente de correlación de Pearson se muestra en la Figura 3.5 donde se ve que el coeficiente mínimo es de 0.995 lo que se traduce en que las tres fases tienen una alta dependencia lineal. En la Figura 3.6 se presenta un cluster jerárquico de las tres fases donde se aprecia que la variable más “independiente” es la de la fase A; nótese que la palabra independiente se pone entre comillas ya que por el coeficiente de correlación de Pearson se sabe que las tres fases tienen una dependencia lineal muy fuerte.

Esto se explica fácilmente desde el punto de vista eléctrico ya que un sistema trifásico de tensiones es equilibrado cuando sus corrientes tienen magnitudes iguales.

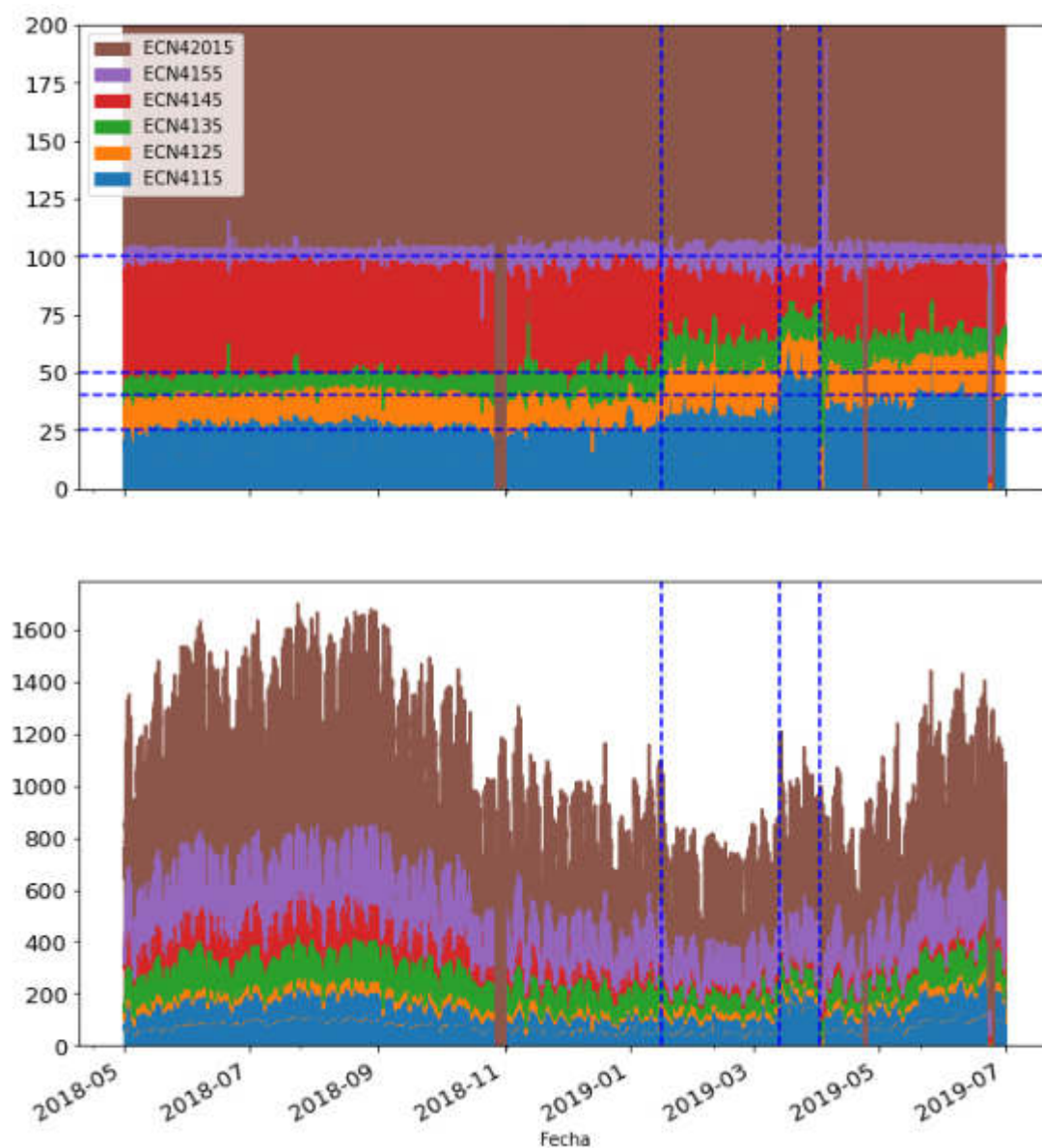


Figura 3.4: Gráficas de área de alimentadores

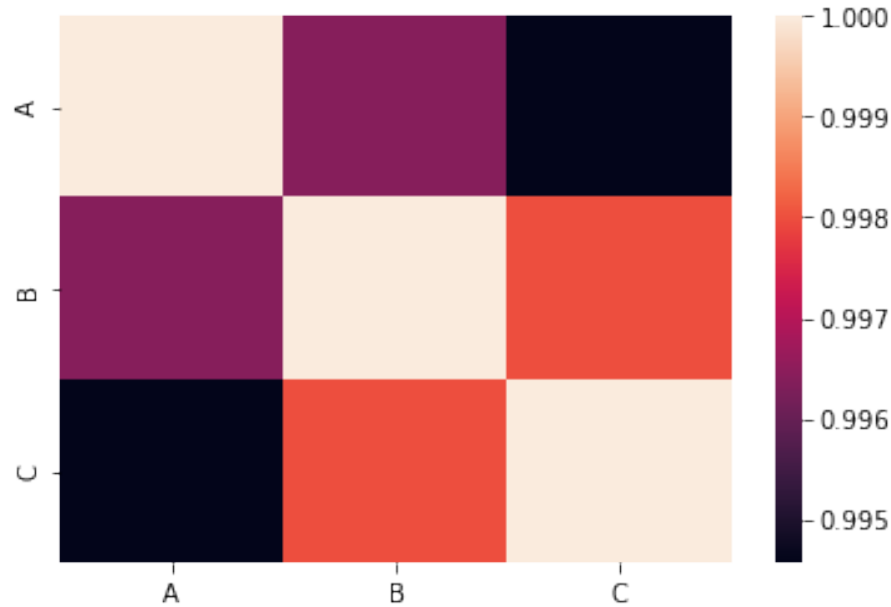


Figura 3.5: Correlaciones de las tres fases

Por lo tanto, es entendible y normal que haya una correlación fuerte entre ellas. Por este motivo se escoge únicamente los datos pertenecientes a la fase A para predecir su demanda.

3.2.3 PRUEBA DE DICKEY-FULLER

La prueba estadística de Dickey-Fuller sirve para saber si una serie de tiempo presenta tendencia o no. Al aplicar la prueba sobre el conjunto de datos se obtiene un p-valor de $1.88e-7$ lo que rechaza la hipótesis nula y se observa que no muestra alguna tendencia, por lo que la serie de tiempo se denomina estacional.

Según datos del INEGI, la población del estado de Nuevo León se ha visto incrementada en 20 % durante los últimos 10 años, hecho que se ve reflejado en el crecimiento de las infraestructuras necesarias para mantener en orden un estado de tal magnitud. El resultado de la prueba Dickey-Fuller sugiere que no existe tendencia en el consumo de electricidad, una posible explicación de este fenómeno es que el intervalo de tiempo es de un año, con lo cual no se puede apreciar dicha tendencia.

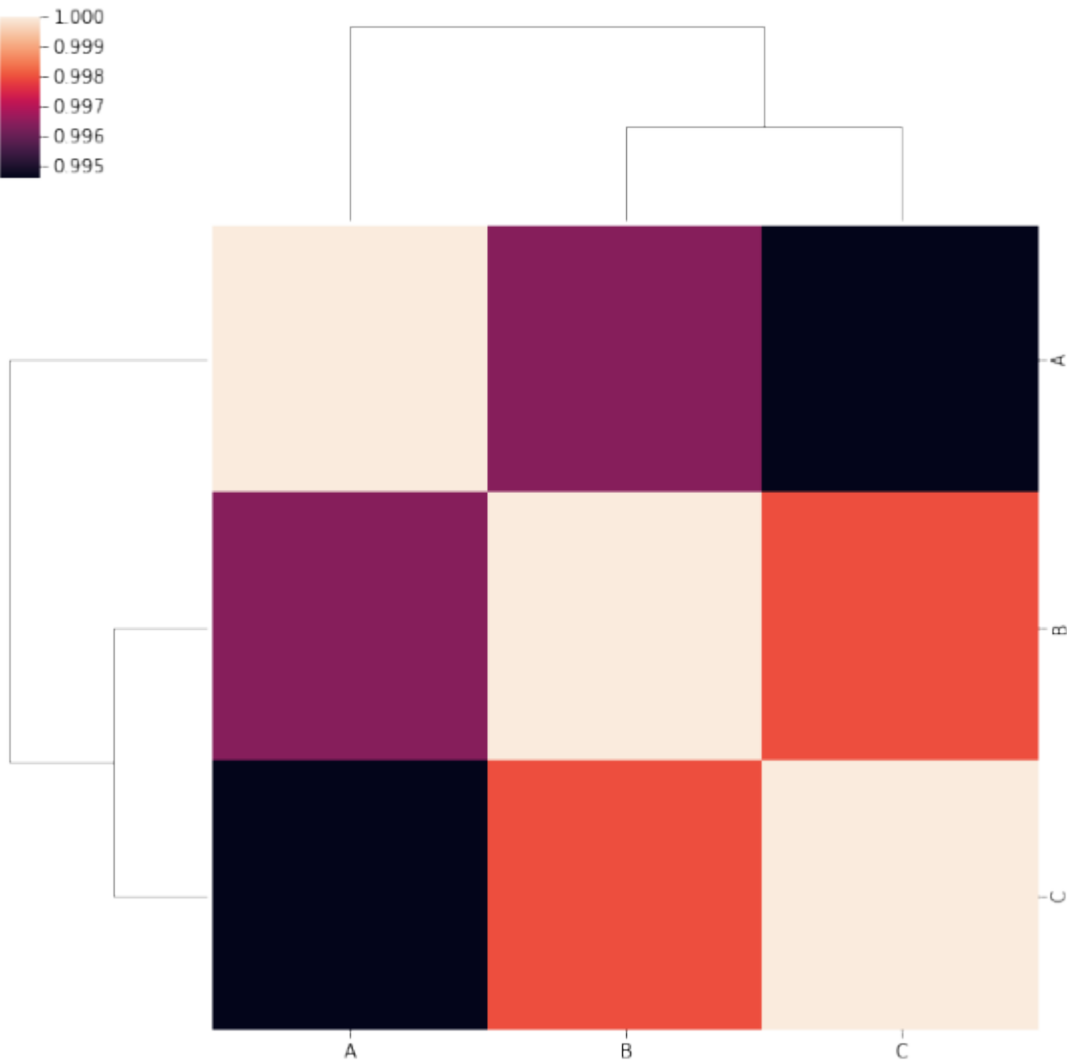


Figura 3.6: Cluster jerárquico de las tres fases

3.3 GRÁFICA POR INTERVALO DE TIEMPO

Para poder observar mejor los comportamientos en ciertos momentos de la serie de tiempo, es necesario graficar intervalos de tiempo más cortos, que no sean gráficas sobre todo el conjunto de datos. En la Figura 3.7 se pueden observar las demandas de los alimentadores en el mes de diciembre donde se ve que hay ciertas mediciones que llegan a ser cero. Al analizar las gráficas por semana, Figura 3.8, se observa un comportamiento atípico entre los días trece y catorce del mes de diciembre, este comportamiento se ve que es atípico porque mientras en algunos alimentadores la demanda llega a cero, en otros alimentadores se ve que la demanda tiene subidas y bajadas más abruptas que los demás días de la semana. Viendo un intervalo de tiempo aún mas corto que la semana, el día, se tiene la Figura 3.9 donde igual se ve que en ciertas horas la demanda es cero.

Estas gráficas sirven de referencia para entender que los datos presentan comportamientos atípicos, que de entrenar los modelos de redes neuronales con estos datos se está forzando a que los modelos encuentren relaciones en datos sucios, por lo que es necesario realizar una limpieza de datos.

3.4 LIMPIEZA DE DATOS

La limpieza de datos es un proceso en el cual se aplican distintas técnicas estadísticas con el fin de detectar comportamientos en los datos que son de una manera distinta a lo que se presenta en la mayoría de los datos registrados. Para detectar estos comportamientos se pueden hacer análisis de manera univariada, es decir, detectar comportamientos anómalos entre la misma variable que se observa o de forma multivariada, lo que significa que se buscan las anomalías considerando todo el conjunto de datos.

Ya que el algoritmo de entrenamiento de redes neuronales consiste en minimi-

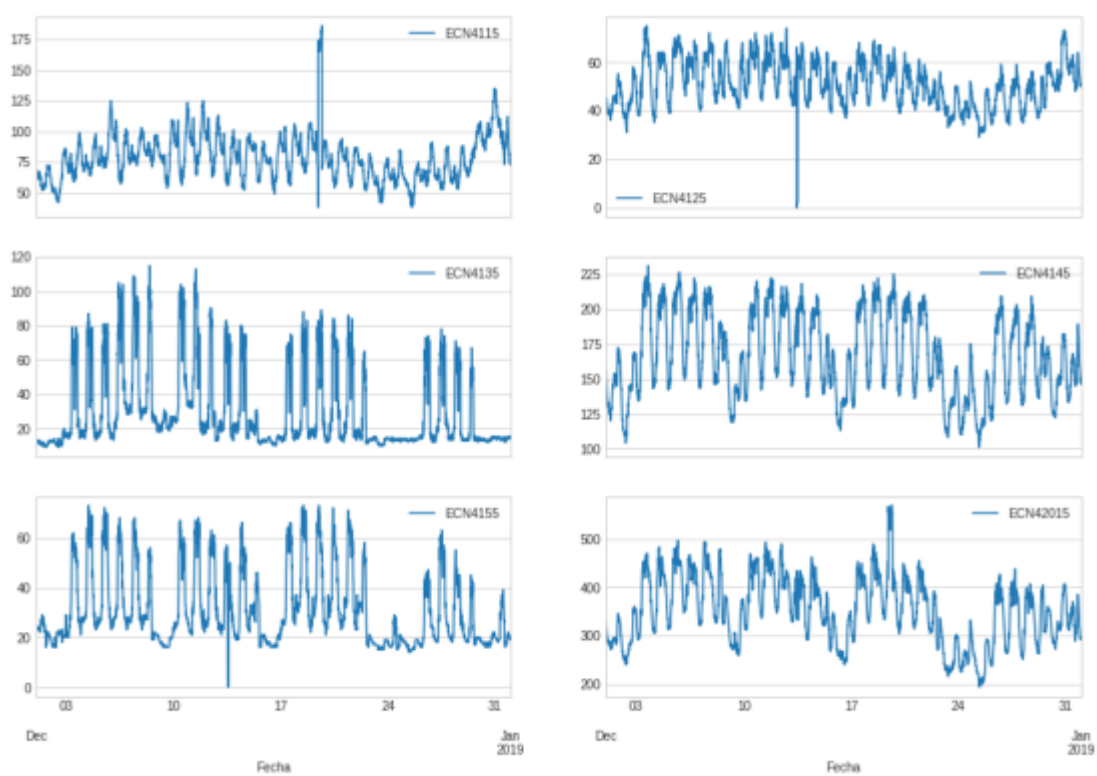


Figura 3.7: Demandas eléctricas en el mes de diciembre.

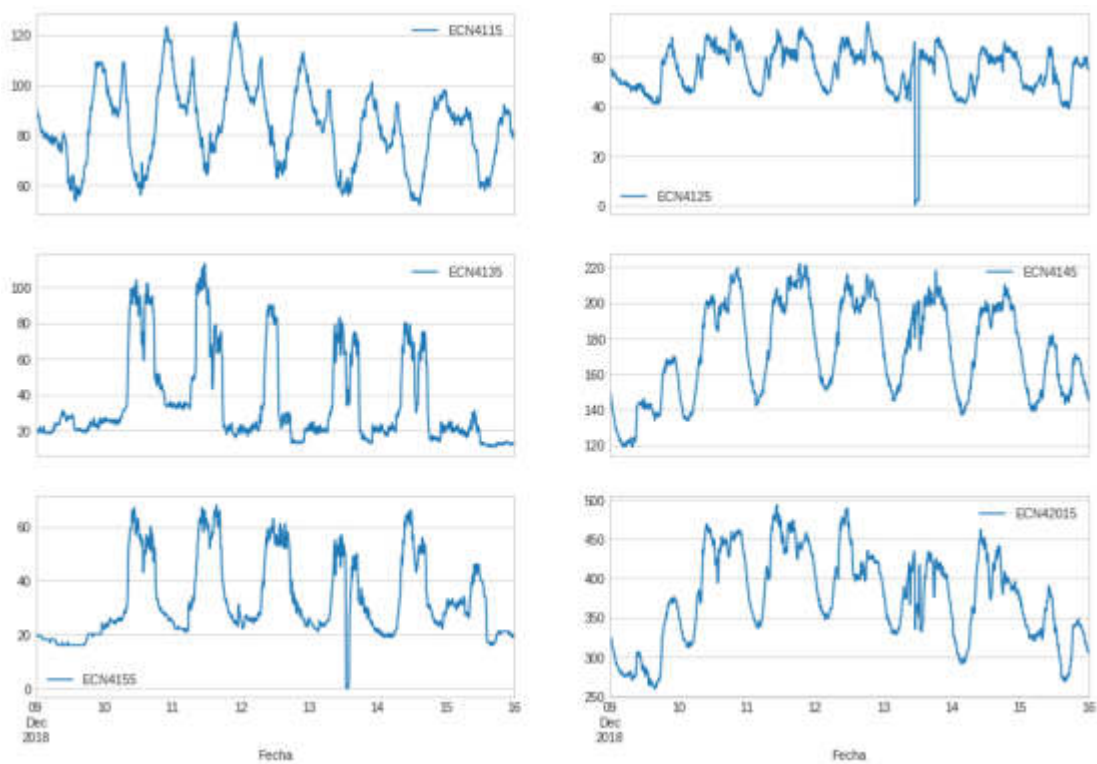


Figura 3.8: Demandas eléctricas en la segunda semana de diciembre.

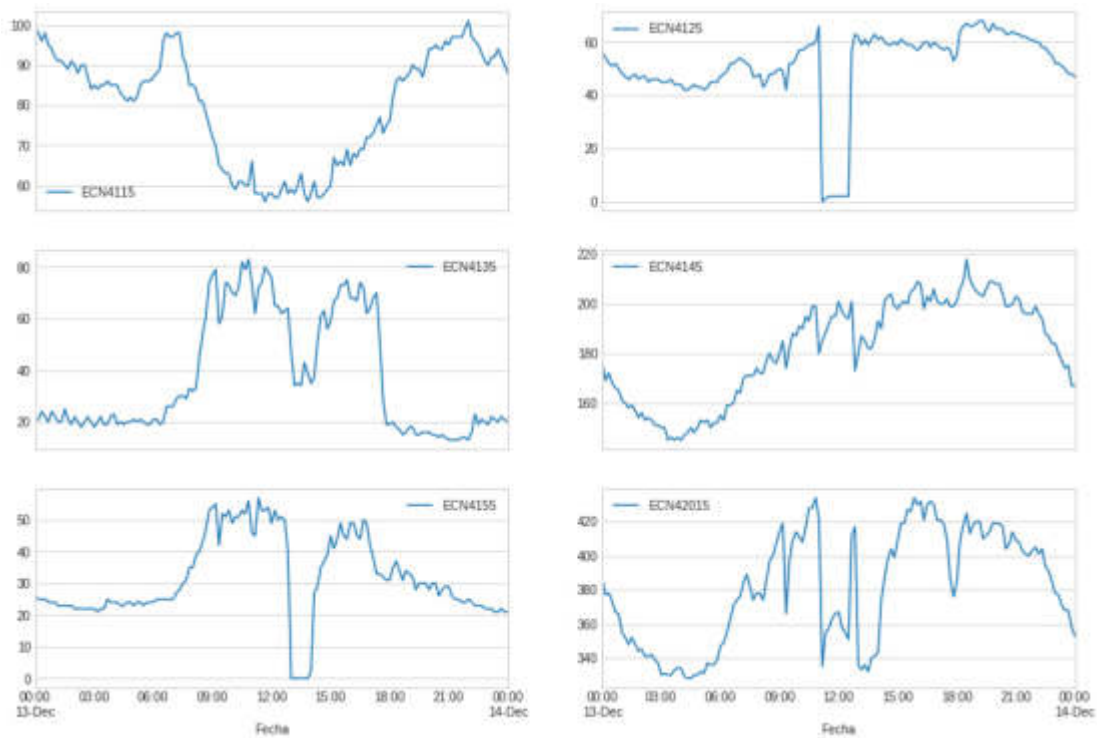


Figura 3.9: Demanda eléctrica el día trece de diciembre.

zar el error entre las predicciones y los datos reales, resulta conveniente hacer una limpieza de datos previas a introducir los datos en los modelos de redes neuronales, de esta manera se ayuda a que el algoritmo de entrenamiento identifique patrones en datos que tienen un comportamiento más estable.

3.4.1 RANGO INTERCUARTIL

El primer método que se aplica a los datos es el de rango intercuartil, este método es univariado, se aplica a cada alimentador y lo que se busca es encontrar los datos que presentan un comportamiento por debajo o por encima de lo que comúnmente sucede. En la Figura 3.10 se muestran las series de tiempo de cada uno de los alimentadores y los datos atípicos se presentan con una franja de distintos colores. Se puede ver que a pesar que hay datos extremos, ya sean superiores o inferiores no en todos los casos son considerados como datos atípicos, y esto es debido

a que el algoritmo está corriendo el rango intercuartil por ventana de tiempo, lo que causa que aunque a simple vista parezcan valores por debajo o por arriba de lo normal el algoritmo no detecta esos datos como atípicos.

3.4.2 LOCAL OUTLIER FACTOR

Este algoritmo permite utilizar datos en múltiples dimensiones por lo que se usa la información de todos los alimentadores simultáneamente. Se analizan los datos atípicos en su conjunto, es decir, en N dimensiones donde N representa la cantidad de los alimentadores que se tienen. Este algoritmo observa la densidad de cada registro con respecto a sus vecinos más cercanos, logrando obtener el 5% de los datos que menos densidad tienen con respecto a sus vecinos, aislando a los registros que presentan más irregularidades con respecto a los demás. Tal y como se muestra en la Figura 3.11 donde se grafican las series de tiempo y con un color rosa se muestran los datos atípicos, nótese que estos datos ocurren en el mismo instante de tiempo para todos los alimentadores.

3.4.3 PCA

Este algoritmo para detectar datos atípicos es multivariado; se aplica una reducción de dimensiones, PCA o análisis de componentes principales por sus siglas en inglés, donde se busca encontrar nuevas variables que sean ortogonales entre sí, que tengan la mayor información (mayor varianza) con respecto a las dimensiones originales y que además la dimensión de estas nuevas variables sea menor que la dimensión original. Una vez encontrada esta función que está en términos de las nuevas variables, los datos originales pasan a la reducción de dimensiones y luego de regreso al espacio original pero solo a una dimensión del espacio original, comúnmente se devuelve a la dimensión con mayor varianza; ahí se encuentra la distancia de los datos originales con respecto a los datos que están reconstruidos, a esto se le llama

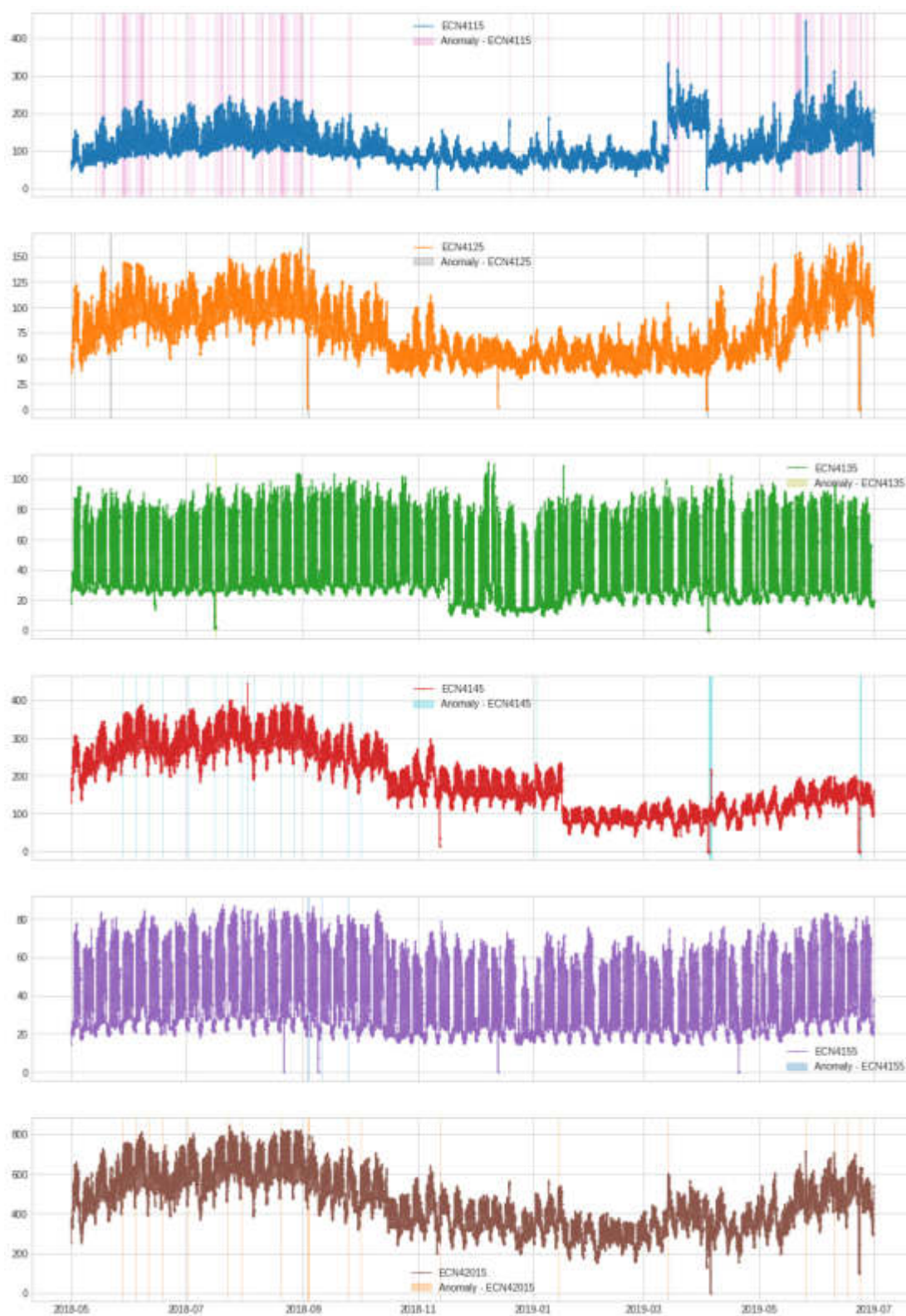


Figura 3.10: Datos atípicos por rango intercuartil.

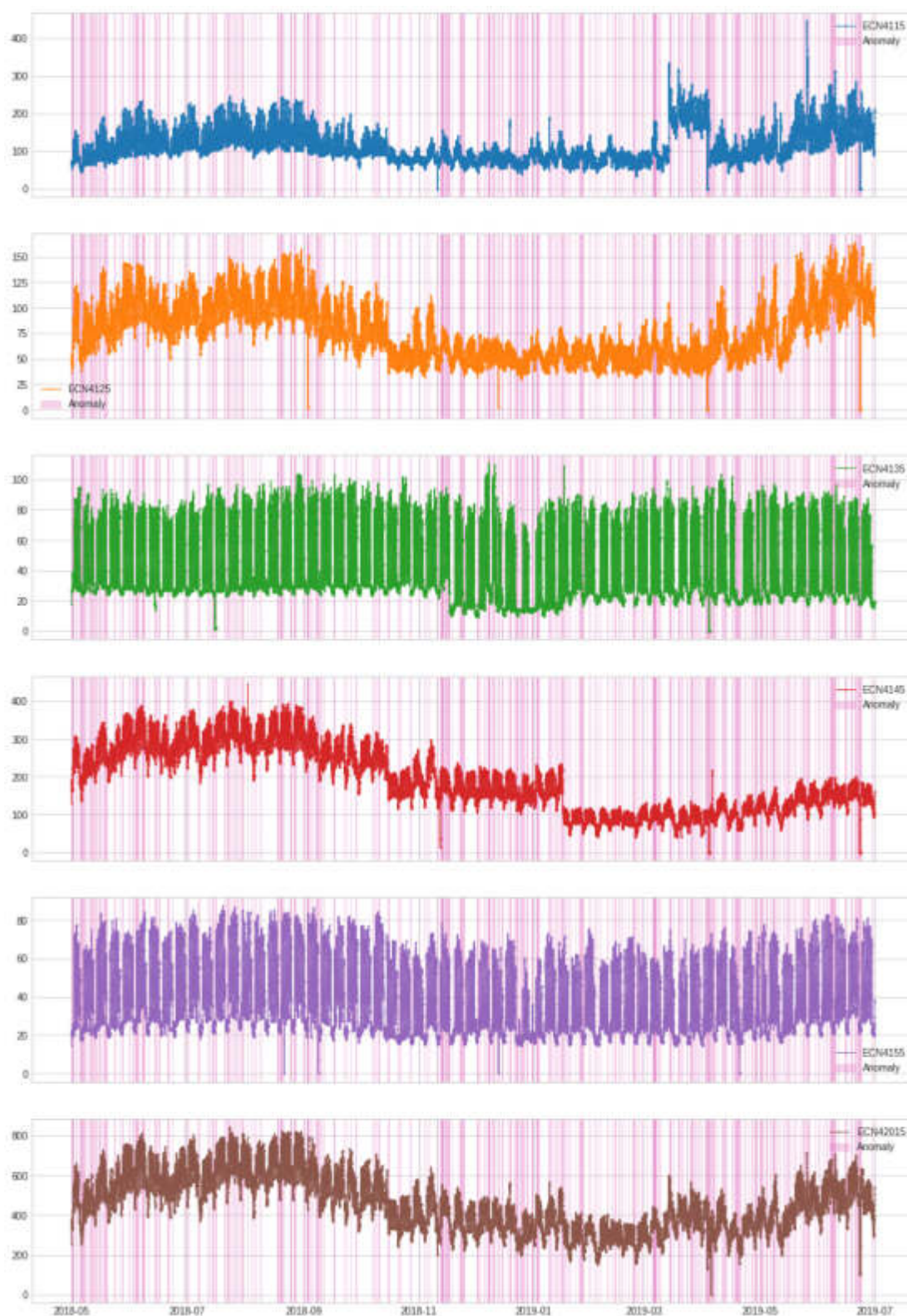


Figura 3.11: Datos atípicos por local outlier factor.

error de reconstrucción, los datos que estén por encima de un límite se consideran datos atípicos. Ya que es un método multivariado, se puede observar en la Figura 3.12 donde los datos atípicos son los mismos para todos los alimentadores.

3.5 ANÁLISIS POSTERIOR A LA LIMPIEZA DE DATOS

Una vez que se han detectado los datos atípicos es posible removerlos y sustituirlos con una interpolación lineal. De esta manera, se pueden relajar algunos comportamientos extraños o con muchos picos en la serie de tiempo convirtiendo los datos originales en una serie más estable. Esto funciona en la práctica porque los valores mínimos y máximos por día o por hora siguen estando ahí.

En la Figura 3.13 se ve la serie de tiempo de la misma semana que en la Figura 3.8 con la diferencia que ya no se tienen los picos inferiores de demanda cero. Así como en la Figura 3.14 donde se ven los datos graficados en el mismo periodo de tiempo que en la Figura 3.9 pero se observa que tiene menos picos.

3.6 DESCOMPOSICIÓN DE SERIES DE TIEMPO

Una serie de tiempo se puede descomponer en tres componentes distintos para conocer su tendencia, su componente estacional y su residual, la finalidad de hacer esta descomposición es permitir a los modelos de redes neuronales conocer estos datos y que encuentren patrones en ellos. Se realizan diferentes descomposiciones considerando ventanas de tiempo distintas, esto es para identificar que ventana de tiempo es la ideal para obtener un error residual que se distribuya normal.

En la gráfica de descomposiciones, Figura 3.15 se ve apreciar que la ventana de tiempo en que los residuos se comportan como normales con mayor significancia, es de 3353 datos hacia atrás, lo cual son 69 días y 20 horas. Además de mandarle los datos de observaciones pasadas a las redes neuronales también

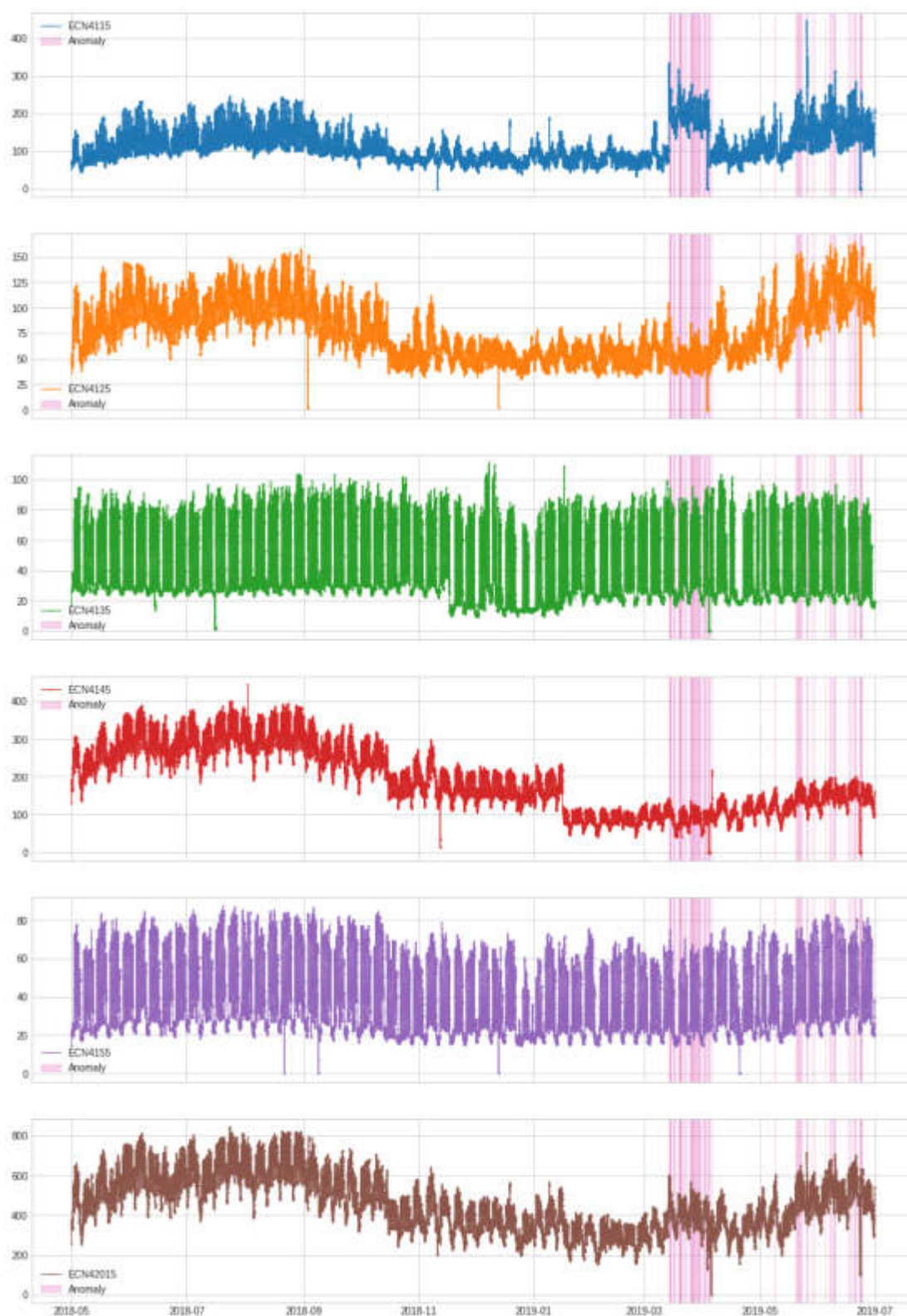


Figura 3.12: Datos atípicos por pca.

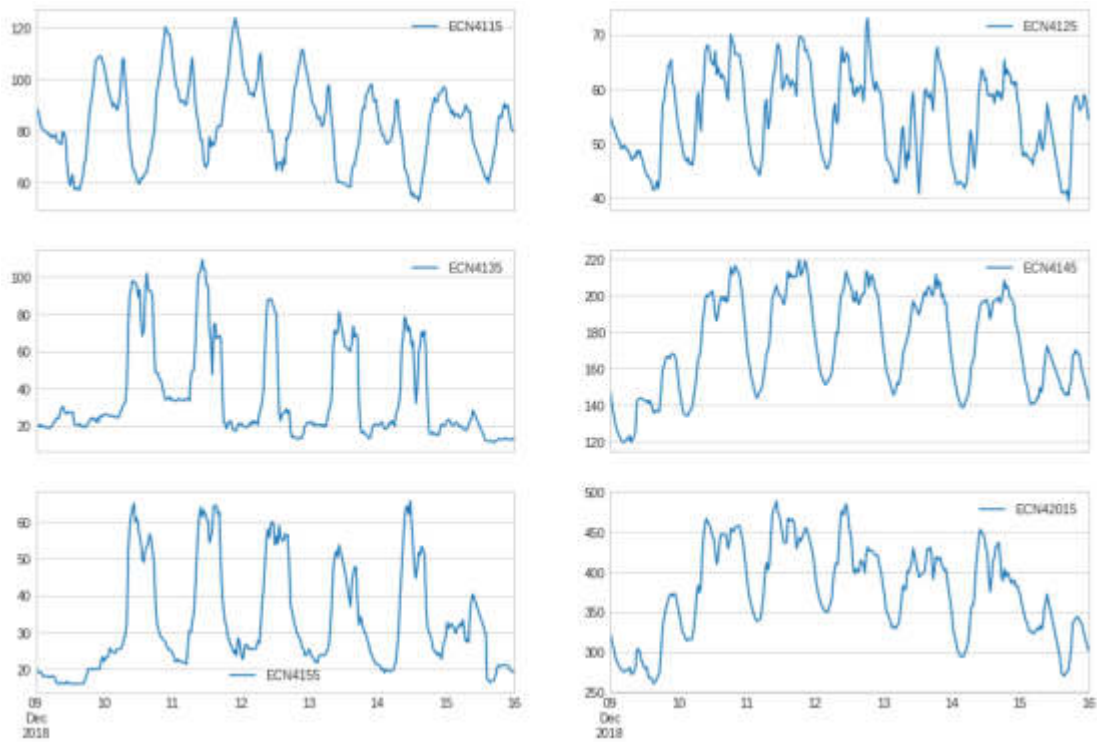


Figura 3.13: Demanda eléctrica por semana con datos reconstruidos.

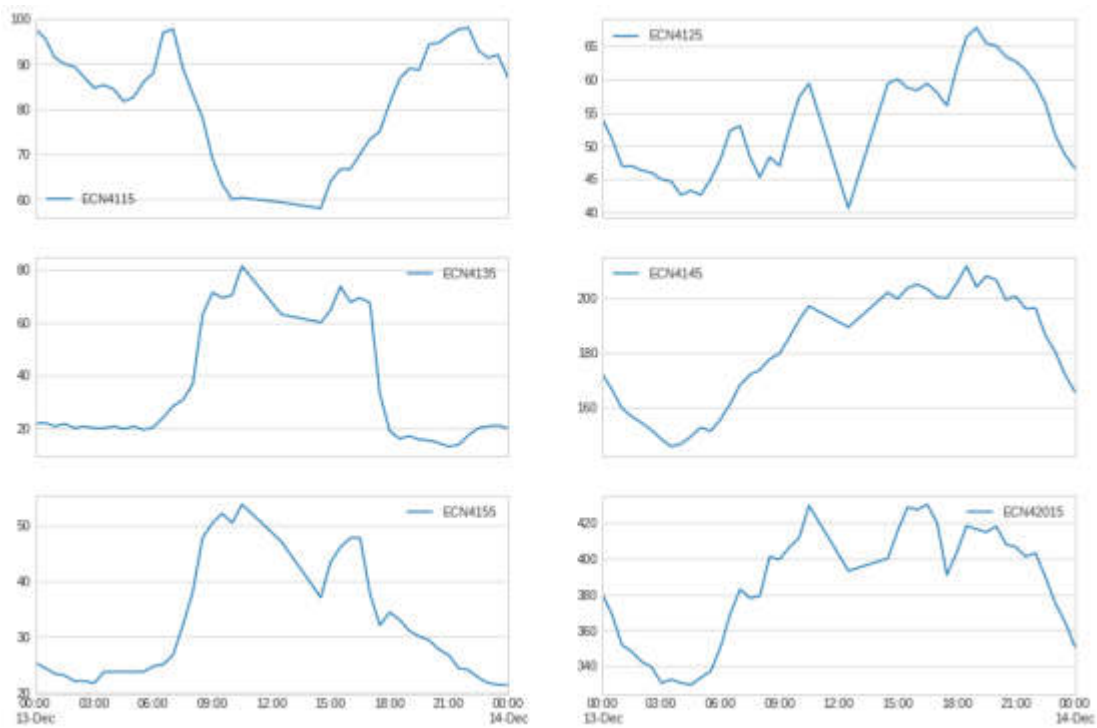


Figura 3.14: Demanda eléctrica por día con datos reconstruidos.

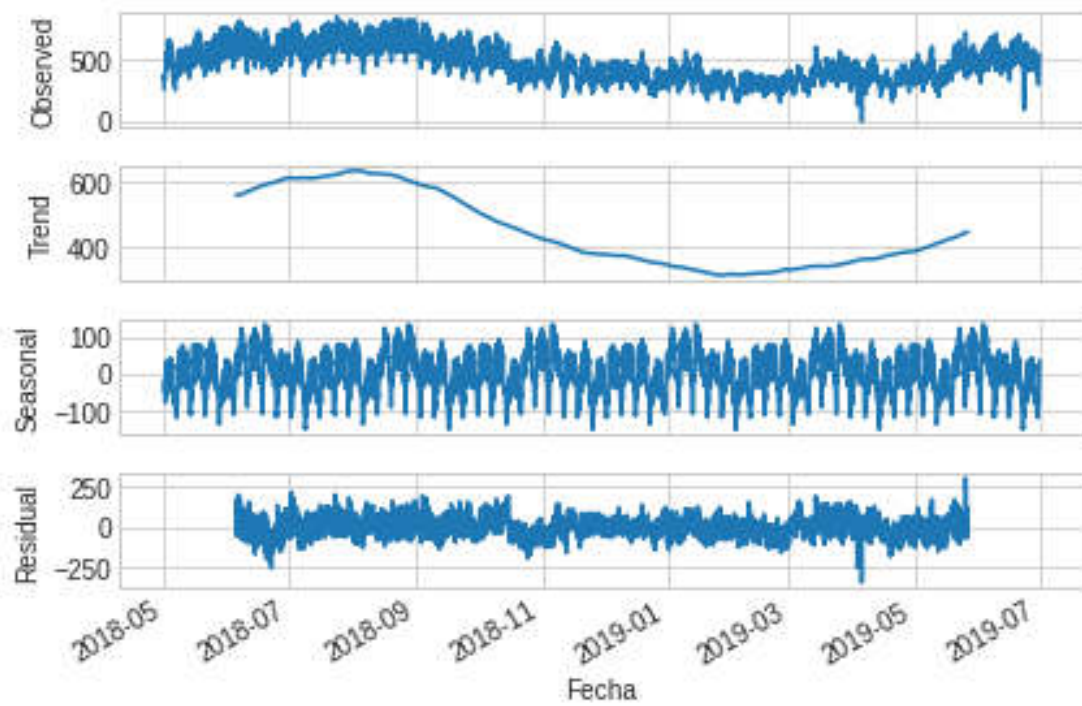


Figura 3.15: Descomposición de la serie de tiempo.

se le muestran los datos de tendencia y estacionalidad obtenidos al aplicar esta descomposición.

CAPÍTULO 4

EXPERIMENTOS Y RESULTADOS

Una vez que se tienen los datos de las demandas eléctricas limpios se procede a diseñar la experimentación para obtener un modelo de redes neuronales que sea capaz de predecir dicha demanda. Hay que mencionar que las demandas que se van a predecir son las de un alimentador padre, esto con el objetivo de delimitar este trabajo. Los experimentos se corrieron en un cluster de Azure Databricks el cual es de un único nodo con 14 gb de memoria ram, 4 núcleos; el tipo de nodo se llama “Standard_DS3_v2” y es de propósito general.

Además de los datos de las demandas se ingresan otro tipo de datos propuestos en la literatura que han probado ser útiles al momento de dar una predicción de demandas eléctricas. En esta sección, se hará la elección de los parámetros fijos y variables escogidos durante los experimentos a la vez que se explican como influyen estos parámetros en los ajustes de los modelos.

4.1 PREPARACIÓN DE LOS DATOS

Con el fin de hacer el entrenamiento de diferentes modelos de redes neuronales es necesario definir los datos que se van a enviar a entrenar, tanto el conjunto de los datos de entrada como el conjunto de los datos de salida. Comúnmente las redes neuronales necesitan conocer la forma de los datos de entrada y de salida ya que

Fecha	original	1_mes	3_meses	tendencia	estacional	residual	dia_0	dia_1	dia_2	dia_3	dia_4	dia_5	dia_6	hora	minutos	semana
2018-08-01 00:00:00	772.666667	632.000000	376.000000	633.645641	78.557180	60.463866	0	0	1	0	0	0	0	0	0	31
2018-08-01 00:30:00	713.333333	636.666667	376.000000	633.668786	64.576115	15.066432	0	0	1	0	0	0	0	0	30	31
2018-08-01 01:00:00	688.000000	636.333333	371.000000	633.737399	70.696707	-16.404106	0	0	1	0	0	0	0	1	0	31
2018-08-01 01:30:00	685.333333	622.333333	360.000000	633.788896	74.758258	-23.213821	0	0	1	0	0	0	0	1	30	31
2018-08-01 02:00:00	672.333333	611.000000	350.333333	633.840690	70.834313	-32.341670	0	0	1	0	0	0	0	2	0	31

Tabla 4.1: Ejemplo de los datos utilizados en los modelos.

al compilar cierta arquitectura de redes neuronales también se están inicializando distintas matrices de datos que es realmente la esencia de las redes neuronales.

Se ha hablado de los datos de demandas eléctricas y su utilidad para ajustar un modelo capaz de predecir demandas eléctricas futuras; en la literatura se han propuesto distintos datos asociados a las demandas que permiten a los modelos tener información extra que permite disminuir el error de las predicciones. Los datos que se usaron en este trabajo fueron las demandas de uno y tres meses anteriores a la medición que se observa, también se utilizó la descomposición de la serie de tiempo en tendencia, estacionalidad y residual. Los datos de salida son las demandas futuras.

Con el objetivo de no solo contar con datos de demandas sino datos que permitan encontrar patrones en el tiempo, se generan columnas binarias que representan a los días de la semana, se tienen siete columnas binarias, además se cuenta con dos columnas adicionales que muestran la hora y los minutos de la medición. Por último, se tiene el número de la semana a la que corresponde la observación. Por lo tanto los registros con los que se entrenan los modelos son como los que se aprecian en la Tabla 4.1.

Es una buena práctica al momento de entrenar modelos de aprendizaje máquina el generar dos conjuntos distintos, el conjunto de entrenamiento y el conjunto de prueba para poder observar si el modelo en verdad está encontrando un patrón en los datos o solo está memorizando ciertos datos que disminuyen el error. El conjunto de entrenamiento se fijó en 70 % de los datos, el conjunto de prueba es del 20 % y un tercer conjunto de validación es del 10 % restante.

La necesidad de predecir la demanda eléctrica es para organizar de una manera más eficiente ciertas reconfiguraciones de la Red General de Distribución, en este

caso de estudio, se requiere que las demandas a predecir sean de al menos siete días en el futuro. En este trabajo, se utilizan las mediciones de ocho días en el pasado para predecir ocho días en el futuro, esto porque en las redes de tipo LSTM se ha demostrado que no tienen buenos desempeños con cantidades de datos mayores de 400 observaciones.

4.2 MODELOS PRELIMINARES

Con los datos de entrada y salida definidos se generan los primeros modelos de redes neuronales. Los modelos de redes neuronales tienen al menos una capa de entrada y una de salida. En este trabajo se exploran dos tipos de modelos, los primeros con una capa intermedia y otros solamente con la capa de entrada y la de salida. Ambos modelos se les aplicó dos escalas distintas a los datos, la escala MinMax y la escala Robusta.

Los experimentos son puestos a prueba variando distintos parámetros de manera aleatoria. Los modelos sin capa intermedia se llevan a cabo cambiando el tamaño del batch, variando el porcentaje del dropout y el número de epochs en que los modelos son entrenados. En la Figura 4.1 se muestran las distribuciones que se siguieron para el experimento sin capa intermedia. Se puede ver la forma aleatoria en la que se eligieron los parámetros.

De la misma manera se ve en la Figura 4.2 la distribución de los parámetros utilizados en el entrenamiento de los modelos con capa intermedia. Además de los parámetros utilizados en los modelos sin capa intermedia, se añaden parámetros del número de neuronas en la capa intermedia y un segundo parámetro de dropout aplicado en la capa intermedia.

De los resultados obtenidos podemos observar en la Figura 4.3 donde se aprecia que no existe diferencia en las escalas utilizadas al entrenar los modelos con respecto al error cuadrado medio, tanto en los modelos con capa intermedia como en los modelos sin capa intermedia. Donde resulta haber una mejoría es en el tiempo de

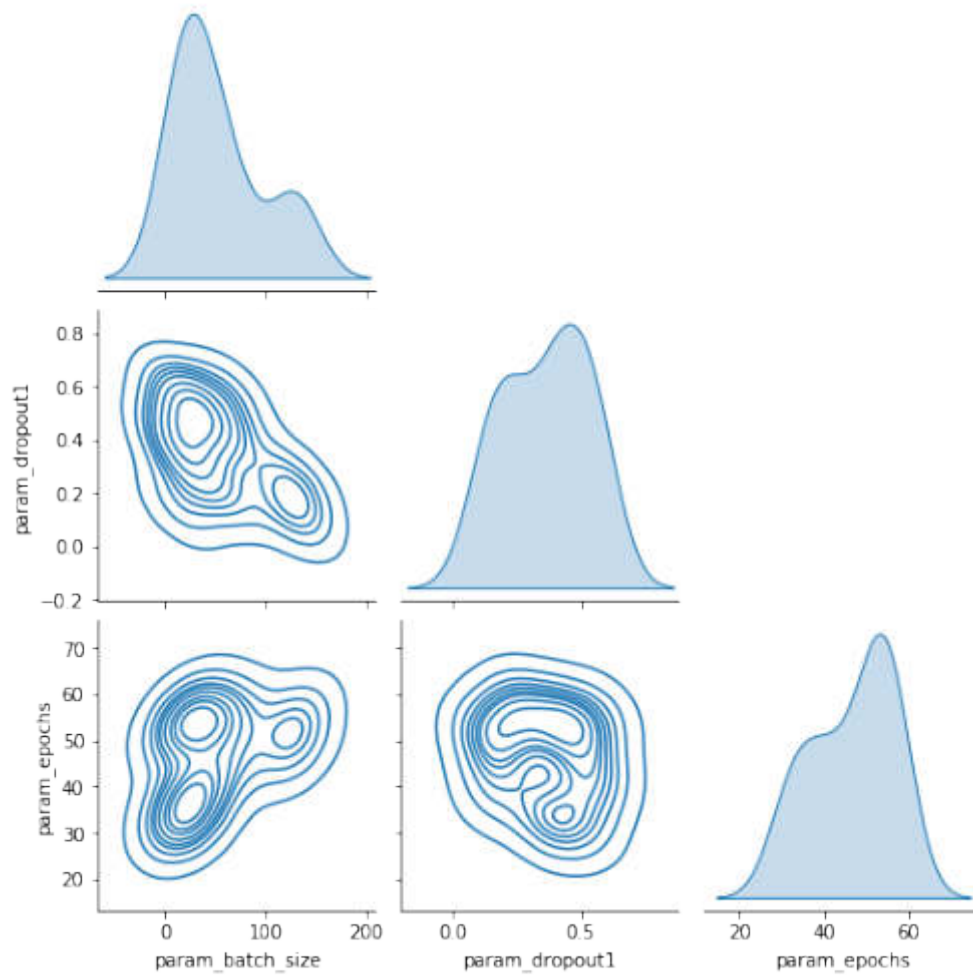


Figura 4.1: Distribuciones de parámetros en modelos sin capa intermedia.

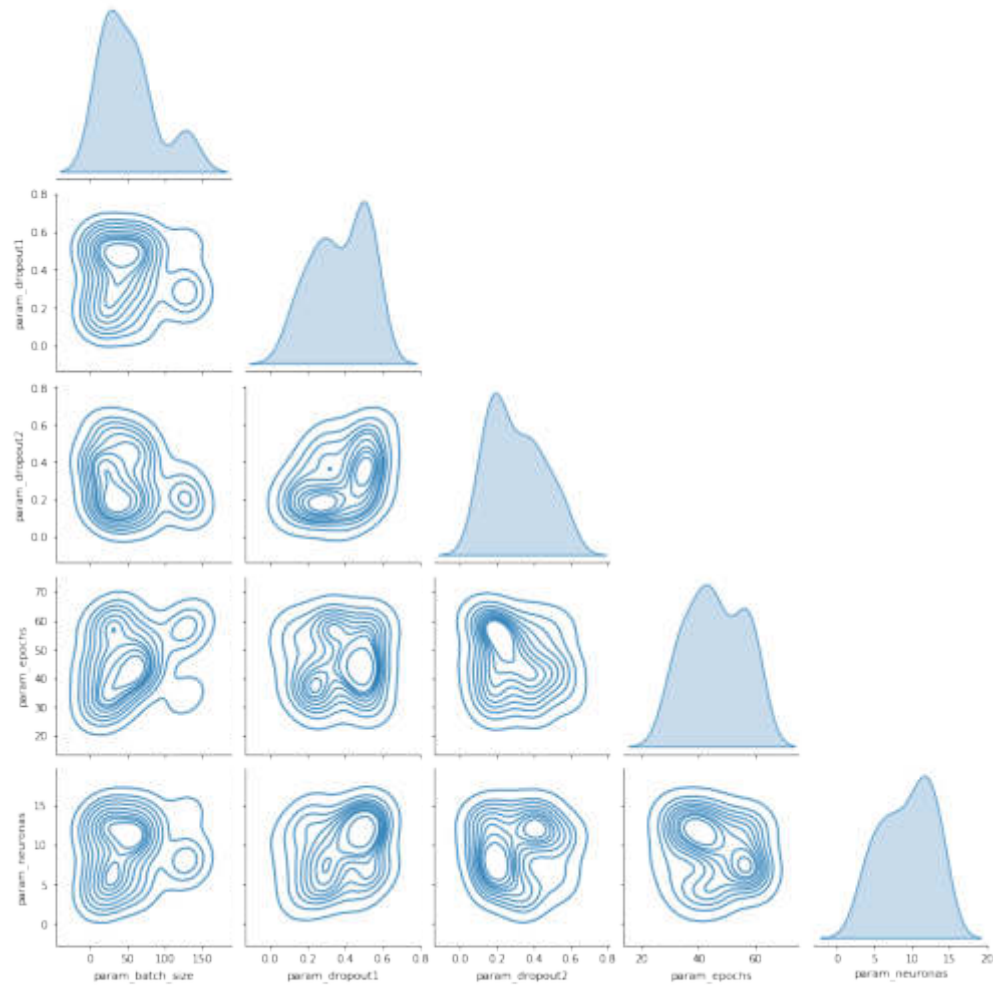
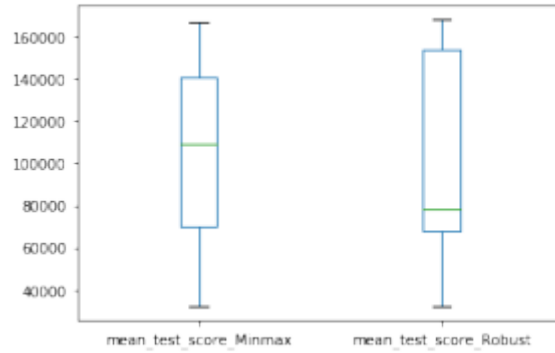
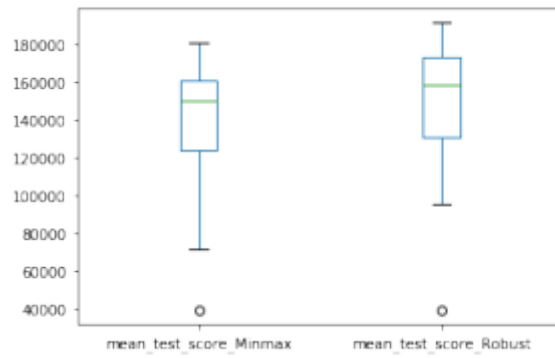


Figura 4.2: Distribuciones de parámetros en modelos con capa intermedia.



(a) Modelos sin capa intermedia



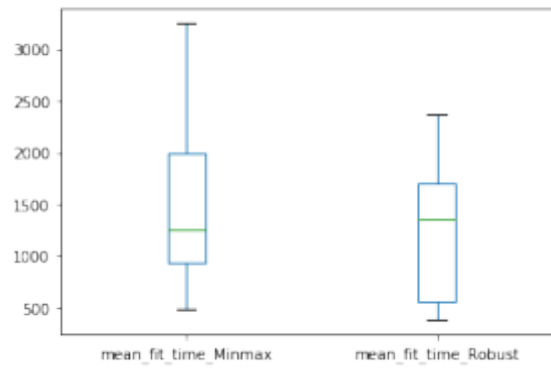
(b) Modelos con capa intermedia

Figura 4.3: Boxplot de error cuadrado medio.

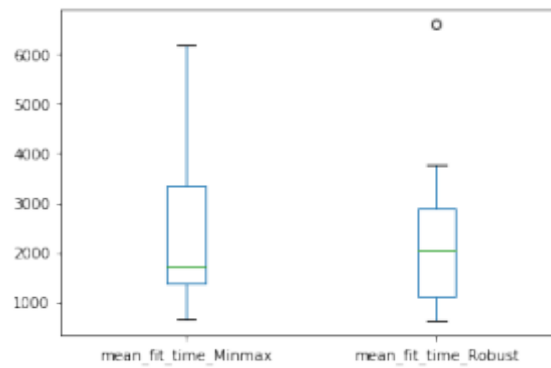
entrenamiento, los modelos entrenados con la escala robusta tienen un menor tiempo de entrenamiento en los dos diferentes modelos, tal como se muestra en la Figura 4.4.

En la Figura 4.5 se muestra el error cuadrado medio de los modelos con las mismas escalas, con capa y sin capa intermedia, donde parece haber una ligera mejoría en los modelos sin capa intermedia. Mientras que hay una clara diferencia en el tiempo de entrenamiento de los modelos sin capa intermedia sobre los modelos con capa intermedia, esto se puede apreciar en la Figura 4.6, donde se entiende que los modelos sin capa intermedia, al tener menos parámetros que entrenar, son modelos más sencillos por lo tanto toman menos tiempo para ser entrenados.

Para analizar los efectos de los parámetros que cambian entre los modelos sin

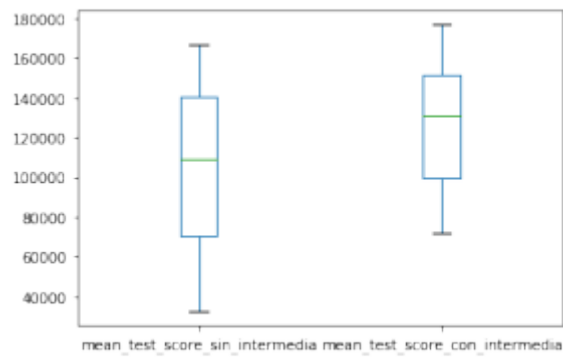


(a) Modelos sin capa intermedia

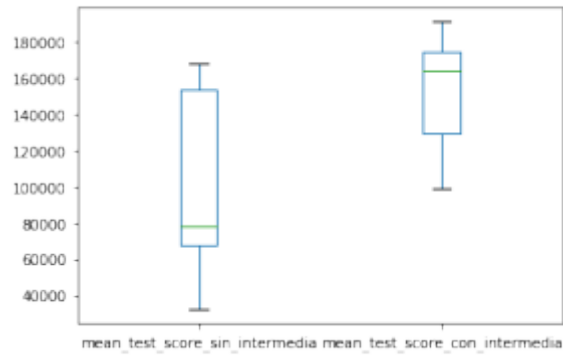


(b) Modelos con capa intermedia

Figura 4.4: Boxplot de tiempos de entrenamiento.

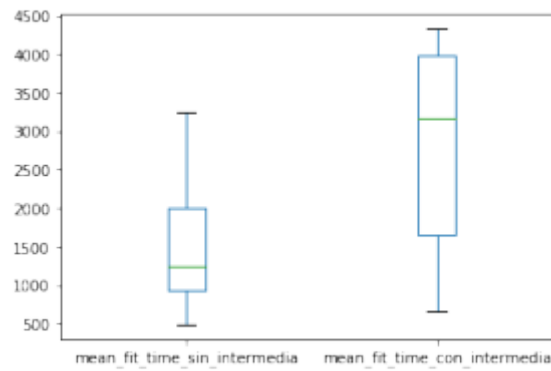


(a) Modelos con escala MinMax

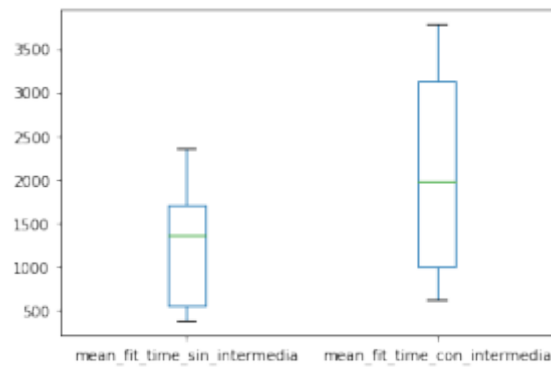


(b) Modelos con escala Robusta

Figura 4.5: Boxplot de error cuadrado medio entre modelos con y sin capa intermedia.



(a) Modelos sin capa intermedia



(b) Modelos con capa intermedia

Figura 4.6: Boxplot de tiempos de entrenamiento entre modelos con y sin capa intermedia.

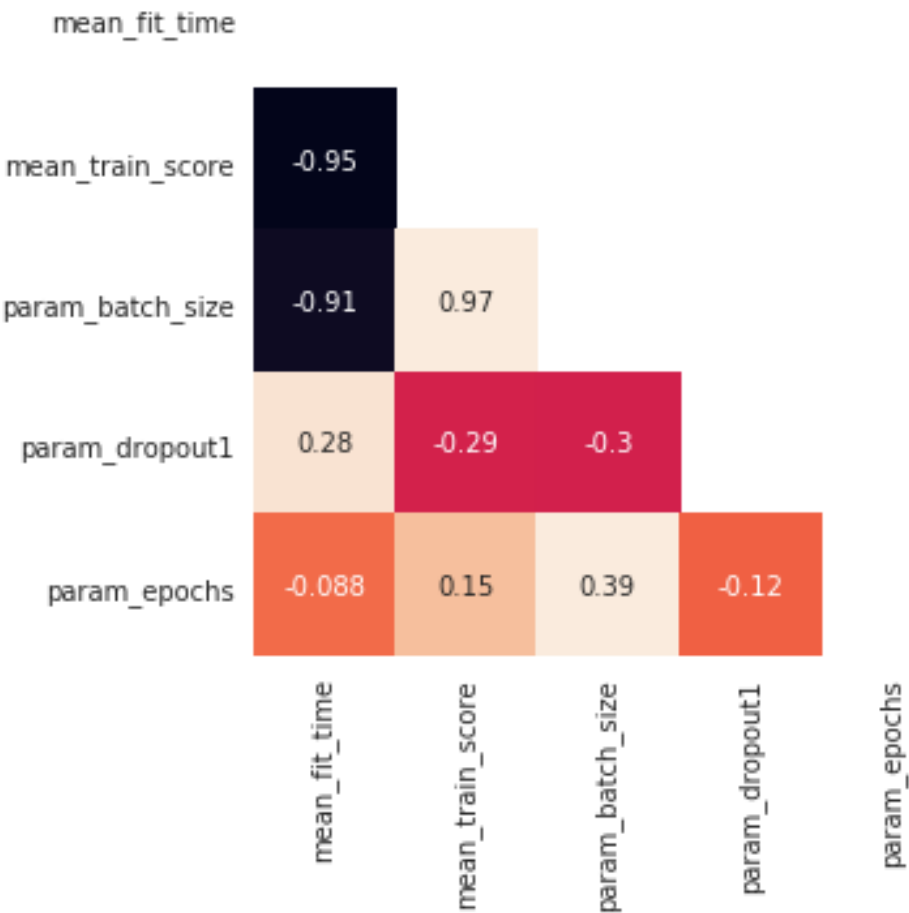


Figura 4.7: Coeficientes de correlación de Spearman en parámetros de modelos sin capa intermedia.

capa intermedia, se elaboró la Figura 4.7 donde se ve la correlación de Spearman existente entre los parámetros y el tiempo de entrenamiento y el error cuadrado medio. Se ve que entre menor tamaño del batch el modelo tarda más en entrenar, lo cual es comprensible porque a menor tamaño del batch se entrena más veces por epoch. También se ve que a menor batch el error cuadrado medio disminuye. La Figura 4.7 presenta las correlaciones de los modelos sin capa que fueron escalados con la técnica minmax, los modelos de escala robusta presentan resultados bastante similares.

Las mismas conclusiones se pueden obtener de la Figura 4.8 donde se encuen-

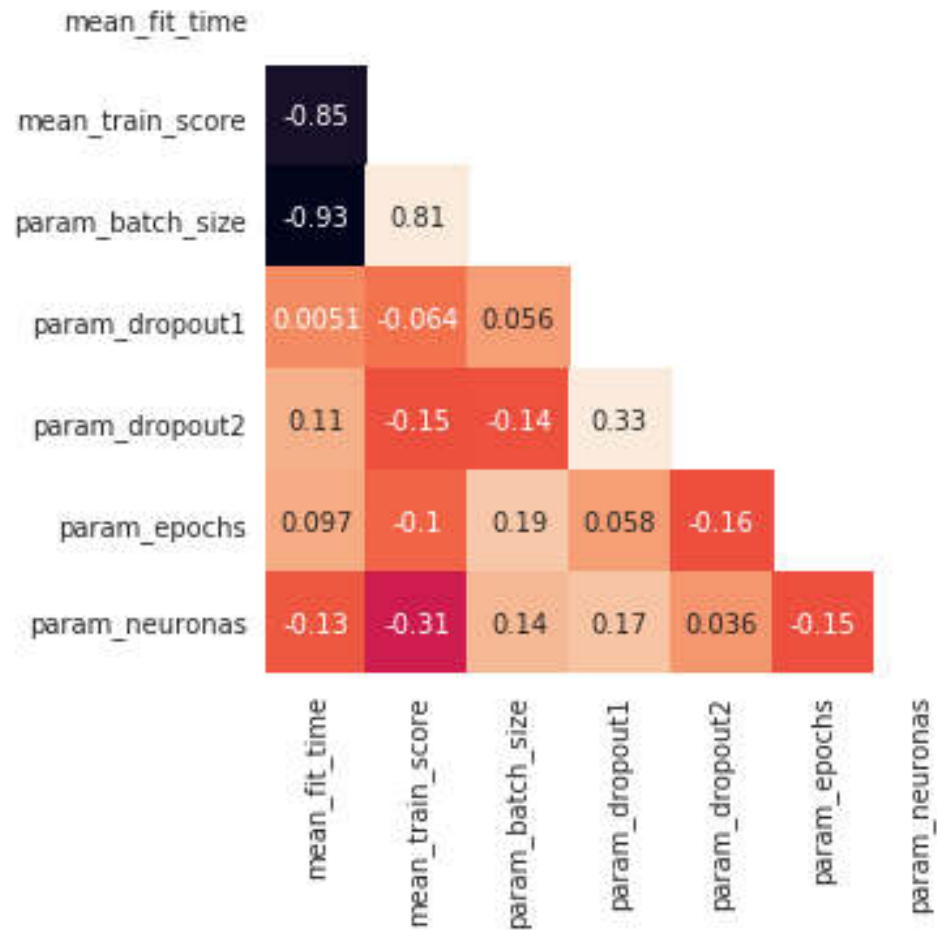


Figura 4.8: Coeficientes de correlación de Spearman en parámetros de modelos sin capa intermedia.

tran las correlaciones de los parámetros con capa intermedia y escala minmax. Otro aspecto a mencionar en estos experimentos es que el parámetro de las neuronas parece no tener relación con el error cuadrado medio.

4.3 MODELOS FINALES

De los resultados obtenidos con los modelos preliminares observó que a menor batch size se obtiene menor error, que la escala robusta presenta ligeras mejoras en

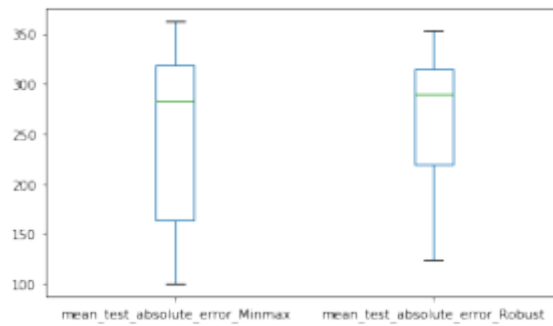
el tiempo de entrenamiento y que los modelos sin capa intermedia tienen tan buenos resultados como los de capa intermedia con la ventaja de que se tarda menos en el tiempo de entrenamiento. Teniendo esto en cuenta, se ajustaron los parámetros del batch con que los modelos son entrenados; los modelos preliminares se entrenaron con tamaños del batch de 16, 32, 64 y 128 mientras que en estos modelos finales se entrenan escogiendo un número de batch al azar entre tres y quince batch por epoch. Otra diferencia entre los modelos preliminares y los modelos finales es que se extendieron los experimentos hechos por cada modelo teniendo mayor aumento de experimentos en los modelos sin capa intermedia.

En la Figura 4.9 es posible ver que no hay diferencia en el tipo de escala, utilizado en los modelos con capa y sin capa intermedia, con respecto al error absoluto medio. Aunque al igual que en los modelos preliminares se sigue presentando una ligera mejoría en el tiempo de entrenamiento de los modelos que utilizan una escala robusta, ver Figura 4.10.

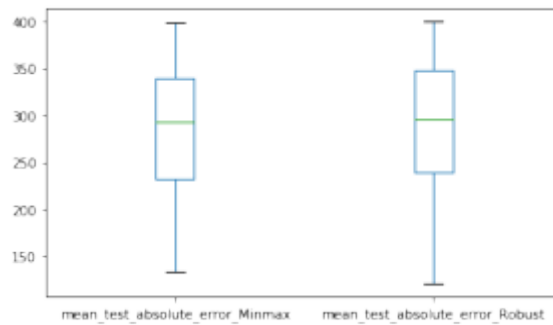
Con respecto a utilizar las mismas escalas en los dos distintos modelos, en la Figura 4.11 se puede notar que no existe diferencia significativa en el error absoluto medio con respecto a los modelos y los dos diferentes tipos de escalas utilizados. En cuanto al tiempo utilizado al momento del entrenamiento sigue habiendo una mejoría en los modelos sin capa intermedia, esto se ve en la Figura 4.12.

Similar a los resultados en los modelos previos se puede observar en la Figura 4.13 que el número del batch tiene una fuerte correlación tanto con el error absoluto medio y el ajuste r^2 en los modelos sin capa intermedia. En los modelos con capa intermedia se observa que el número de neuronas en la capa intermedia influye en una mejora en el desempeño de los modelos con respecto al ajuste r^2 y el error absoluto medio, junto con el parámetro del tamaño del batch como se muestra en la Figura 4.14.

Analizando los resultados se ve que el modelo que genera un menor error absoluto promedio es el que se le aplica una escala MinMax, el que no contiene capa intermedia, es decir, solo tiene la capa de entrada y la capa de salida, que tiene

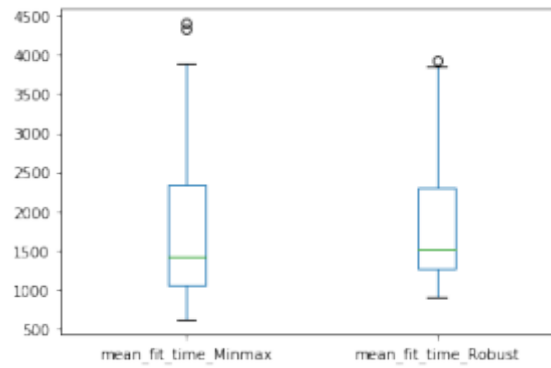


(a) Modelos sin capa intermedia

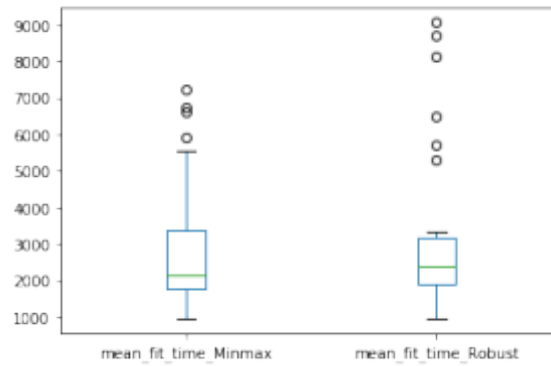


(b) Modelos con capa intermedia

Figura 4.9: Boxplot de error absoluto medio.

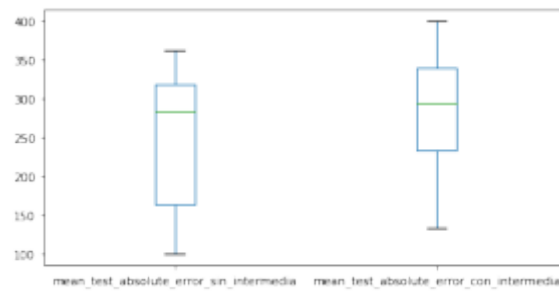


(a) Modelos sin capa intermedia

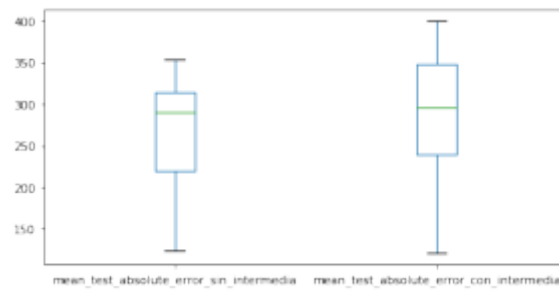


(b) Modelos con capa intermedia

Figura 4.10: Boxplot de tiempo de entrenamientos.

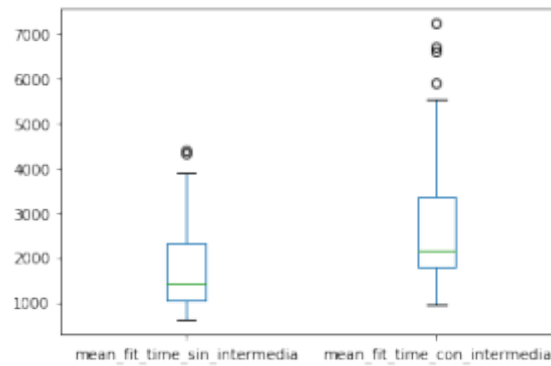


(a) Modelos con escala MinMax

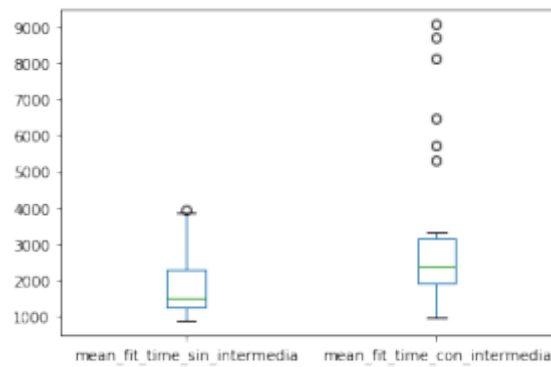


(b) Modelos con escala Robusta

Figura 4.11: Boxplot de error absoluto medio entre modelos con y sin capa intermedia.



(a) Modelos sin capa intermedia



(b) Modelos con capa intermedia

Figura 4.12: Boxplot de tiempos de entrenamiento entre modelos con y sin capa intermedia.

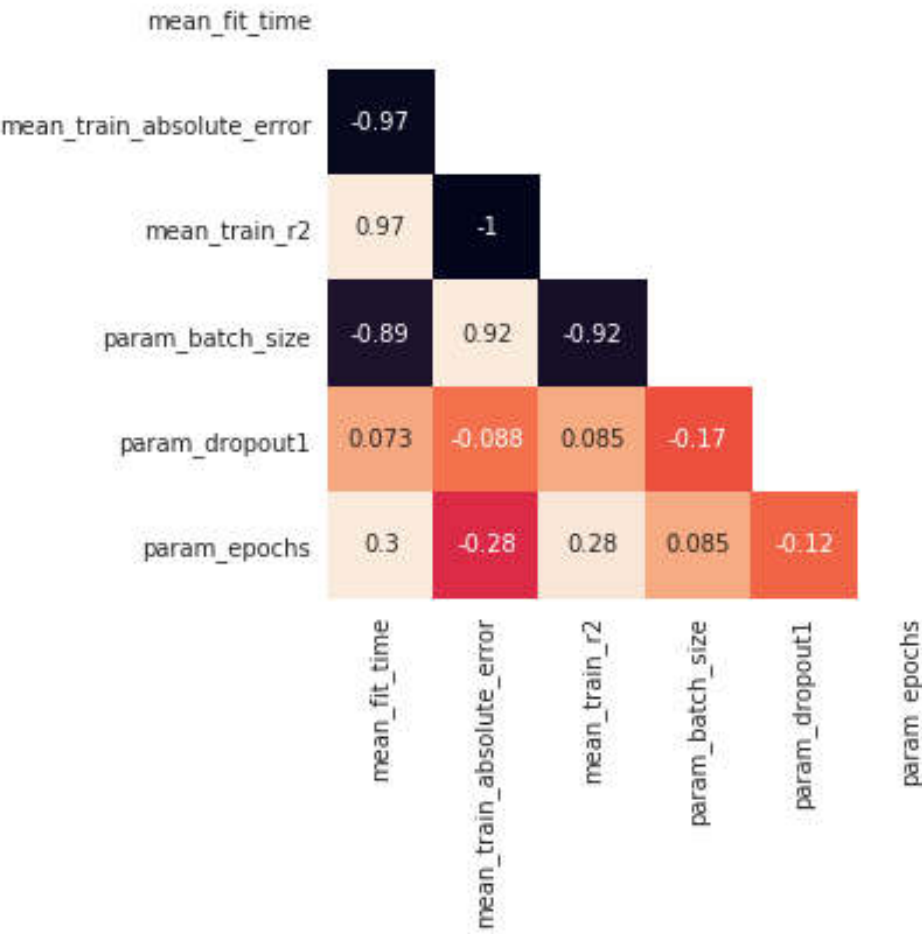


Figura 4.13: Coeficientes de correlación de Spearman en parámetros de modelos sin capa intermedia.

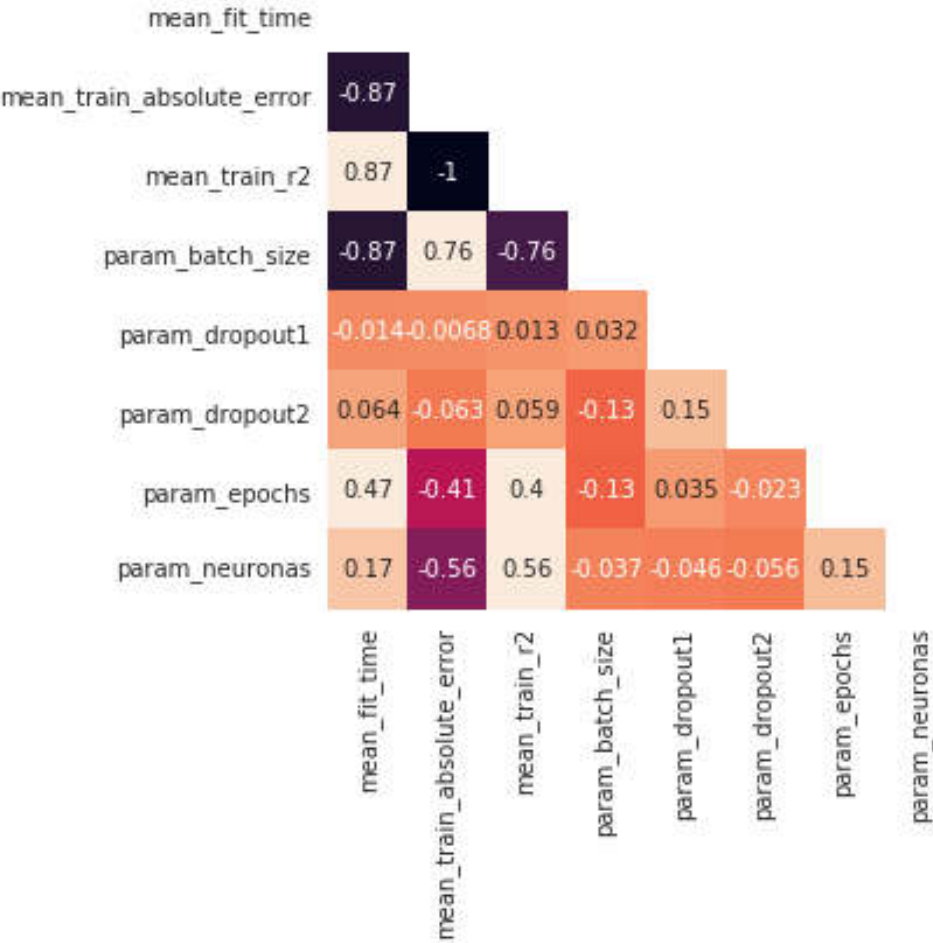
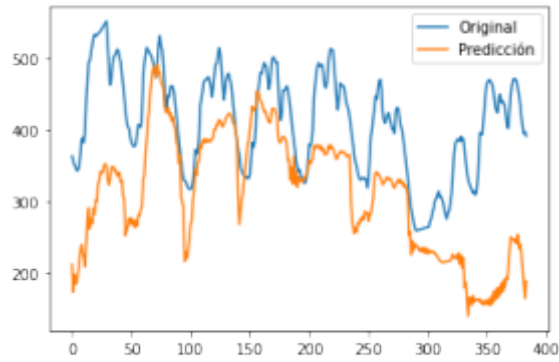
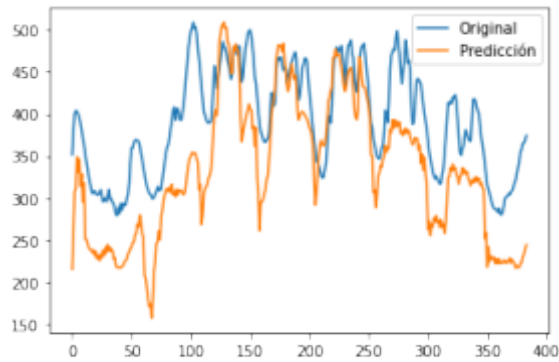


Figura 4.14: Coeficientes de correlación de Spearman en parámetros de modelos sin capa intermedia.

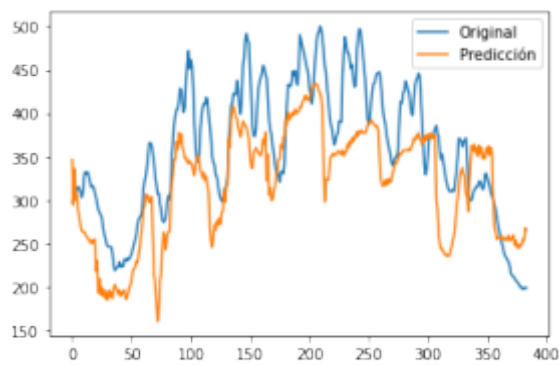
un tamaño de batch en el entrenamiento de 9, con 54 epochs. En la Figura 4.15 se muestran algunos ejemplos de las predicciones hechas por el modelo que presenta mejor desempeño.



(a) Ejemplo 1



(b) Ejemplo 2



(c) Ejemplo 3

Figura 4.15: Ejemplos de predicciones del modelo ganador.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

En los experimentos realizados se ve que no existe una diferencia significativa entre los modelos con capa intermedia y sin capa intermedia, por lo que se toma como mejor modelo el que no tiene capa intermedia ya que es más rápido en el tiempo de entrenamiento. Los experimentos con los modelos preliminares sugieren que los modelos con escala robusta tienen una ligera ventaja en el tiempo de entrenamiento, sin embargo el menor error absoluto medio presentado en los modelos sin capa intermedia se presenta al utilizar el tipo de escala MinMax.

Los experimentos han demostrado que es posible obtener un error absoluto medio de 100 Kw/h en los modelos con una sola capa de entrada, quitando el 10 % de la información de las neuronas en cada batch, lo cual es el parámetro de dropout, y entrenando el modelo con un tamaño de batch de nueve.

5.1 RECOMENDACIONES

Una recomendación es contar con un mayor rango de tiempo en los datos, en lugar de tener un año que se tenga la información de varios años. Esto porque al aplicar ciertas transformaciones se quitaron cierto volumen de los datos.

En la literatura se ha visto que un dato que resulta de utilidad al predecir la demanda eléctrica es el de la temperatura. Una manera de mejorar los resultados

obtenidos es tener las demandas mínimas y máximas por día para dar más información de calidad y que los modelos sean capaces de encontrar relaciones entre estos datos.

5.2 TRABAJO A FUTURO

Este trabajo puede presentar una mejora en las métricas si en lugar de seguir la recomendación de predecir por lo menos siete días futuros, se realice un estudio para determinar cuál es la ventana de tiempo mas adecuada para la predicción. El mismo concepto se puede utilizar para los datos que se utilizan en el pasado para predecir, también es posible realizar un estudio para determinar la cantidad de datos históricos que se deben considerar para realizar la predicción.

Para dar continuidad a este trabajo, se podrían tomar como base los modelos con mejor desempeño, en todos los alimentadores. Incluso cambiar la periodicidad de los datos a predecir.

APÉNDICE A

APÉNDICE

A.1 DICKEY-FULLER

La prueba Dickey-Fuller sirve para determinar si la serie de tiempo presenta tendencia o no. Ajusta un modelo autorregresivo y examina la hipótesis nula de que la serie presenta tendencia contra la alternativa de que el modelo autorregresivo sea estacionario Cheung y Lai (1995).

A.2 DESCOMPOSICIÓN DE SERIE DE TIEMPO

Una serie de tiempo es una variable aleatoria cuyas mediciones están enlazadas entre si por un factor de tiempo. Muchos fenómenos de la vida cotidiana pueden ser vistos como series temporales, por ejemplo el incremento en la venta de los supermercados cada quincena o el alza de los precios año tras año. Se han encontrado distintas metodologías para analizar este tipo de datos permitiendo conocerlos de manera tal que se pueden encontrar relaciones entre las observaciones, hecho importante para poder predecirlos.

Supóngase que se está midiendo la cantidad de clientes que asisten a un centro comercial de una ciudad, es probable que los fines de semana haya más afluencia

que entre semana; este es un patrón de estacionalidad. La tendencia es la variable de clientes si se registra un promedio por mes a través de los años. El residuo o componente aleatorio es como los registros altos en los centros comerciales en días de partido o en días festivos.

Entonces una serie de tiempo se puede descomponer en tres elementos distintos:

- Estacionalidad
- Tendencia
- Componente aleatorio

A.3 NÚMERO DE CAPAS EN LAS REDES NEURONALES

Cada capa de una red neuronal está compuesta por neuronas y la neurona por dentro tiene distintos parámetros por lo que es lógico que al incrementar el número de capas es probable que también se aumenten la cantidad de parámetros que hay que ajustar. Después del procesamiento de datos quedan poco más de 13 mil registros por lo que no se pueden implementar redes neuronales que tengan más parámetros que esa cantidad de datos por lo tanto se considera hacer técnicas de Machine Learning en lugar de técnicas de Deep Learning. Lo anterior también está motivado por distintos autores que obtuvieron buenos resultados con modelos de una sola capa oculta aparte de tener la ventaja que éstos necesitan menos cantidad de datos para su entrenamiento.

Esto da lugar a que se utilicen dos tipos de arquitecturas diferentes:

- Capa de entrada, capa oculta y capa de salida
- Capa de entrada y capa de salida

A.4 TIPO DE ESCALA

Las redes neuronales son sensibles a las escalas esto por las funciones de activación que se utilizan como la tangencial y sigmoide. Existen distintos métodos para cambiar de escala, en lo que consisten estos métodos es en aplicar una función a los datos para que cambien el rango original, por lo que hay que ser cuidadosos al utilizar ciertas transformaciones ya que pueden depender de ciertos supuestos que de no cumplirlos hacen que los datos pueden alterar su distribución. Para transformar los datos se usaron dos métodos distintos:

- Escala MinMax: Esta es un tipo de escala en la cual no es necesario que los datos tengan cierta distribución. La fórmula matemática de esta escala es $x_{scaled} = (x - \min(x)) / (\max(x) - \min(x))$ dejando de esta manera a los datos escalados en el rango $[0, 1]$ pero los datos se comportan exactamente igual a los originales solo que en un espacio distinto.
- Escala Robusta: Al aplicar este tipo de escala si se puede ver afectada la distribución de los datos. En la fórmula de la escala minmax se puede ver que depende de los valores máximos y mínimos de los datos, por lo cual en presencia de datos atípicos puede hacer que la mayoría de los datos pasen a un espacio muy pequeño, por ejemplo $[0.2, 0.6]$ y aunque se sabe que entre dos números reales siempre existe otro número real por lo cual cada datos estaría bien representado con algún valor real, la realidad es que se trabaja en ambientes computacionales por lo que si muchos datos entran en ese conjunto tan pequeño se corre el riesgo de que no se pueda representar bien cada dato con respecto a la cercanía o lejanía de sus vecinos.

La fórmula de la escala robusta es $x_{scaled} = (x - q1(x)) / (q3(x) - q1(x))$ donde $q1(x)$ es el dato más pequeño del primer intercuartil y $q3(x)$ es el dato mas grande del tercer intercuartil. De esta manera se acercan los datos atípicos al grueso de la población.

A.5 HIPERPARÁMETROS EN LAS REDES NEURONALES ARTIFICIALES

A.5.1 EPOCHS

Al entrenar redes neuronales es necesario mostrar todo el conjunto de datos varias veces con el objetivo de que el algoritmo que se esté utilizando para minimizar el error ajuste los pesos necesarios para modelar el conjunto de datos en cuestión. Una epoca consiste en mostrar el conjunto de datos a la red neuronal, por lo que el número de épocas es el número de veces que se utilizan todos los datos para entrenar el modelo.

Mostrando pocas veces el conjunto de datos es posible que los pesos no se ajusten de una manera eficiente lo que provoca el problema del poco entrenamiento, creando un mal desempeño en las predicciones de las redes neuronales. Por otra parte, al mostrar muchas veces el conjunto de datos al modelo de red neuronal puede causar que el modelo tenga un sobreajuste y por lo tanto se "aprenda" los datos. Este parámetro es importante porque influye de manera directa en el resultado del modelo.

A.5.2 DROPOUT

La técnica dropout es una técnica que ayuda a prevenir el sobreajuste de las redes neuronales Baldi y Sadowski (2013). Esta técnica consiste en que durante el entrenamiento ciertas neuronas se dejan fuera de la actualización de los pesos, esto obliga a las neuronas a ser robustas al no depender de la actividad de otras neuronas en específico.

A.5.3 TAMAÑO DEL BATCH

En el entrenamiento de las redes neuronales los datos son pasados un determinado número de epochs y durante cada epoch se ajustan los pesos cada que pasan ciertos números de observaciones o muestras, a este número se le llama tamaño del batch.

A.6 FUNCIONES DE PÉRDIDA

Una definición sencilla de una función de pérdida puede ser de la siguiente manera: Dada una predicción y un valor real, la función de pérdida mide la diferencia entre el valor reportado por el algoritmo o modelo de predicción y la salida deseada.

La estimación de los parámetros del modelo depende en gran medida de la utilización de diferentes funciones de pérdida. Cada una de estas funciones se soportan en un estadístico por ejemplo la media, la media al cuadrado o la mediana, el cual se convierte en un criterio importante para la decisión del modelo de estimación. Son precisamente las medidas las que condicionan el cálculo matemático de la función misma para la evaluación de modelo de pronóstico. Se utilizan diferentes métricas que se describen a continuación.

A.6.1 ERROR PORCENTUAL ABSOLUTO MEDIO

El error porcentual absoluto medio mide el tamaño del error en términos porcentuales lo cual lo hace no dependiente de la escala. Se calcula como el promedio de los errores porcentuales sin tener en cuenta el signo.

A.6.2 ERROR ABSOLUTO MEDIO

Se define el error absoluto medio como la magnitud promedio de los errores de un ejercicio de pronóstico sin tener en cuenta su signo, es decir, el promedio de los valores absolutos de los errores calculados.

Algunas de las características del error medio absoluto es que el peso asignado a cada uno de los errores calculados es el mismo; el uso de esta medida es aconsejable para describir errores uniformemente distribuidos; el hecho mismo de ser una medida absoluta podría convertirse en una desventaja, sin embargo, por no ser de naturaleza cuadrática, recibe menos influencia de valores atípicos.

En las ventajas se encuentran que la mayoría de modelos tienen un supuesto de que los errores se encuentran distribuidos como una normal y no como una función uniforme, es posible encontrar métricas que se ajustan de una mejor manera a los datos.

Un importante aspecto en las métricas de error es su capacidad de discriminar entre los resultados del modelo. Con esta métrica los resultados podrían estar afectados por una gran cantidad de valores de error promedio sin necesariamente reflejar errores de gran tamaño.

A.6.3 ERROR CUADRADO MEDIO

Esta medida es dependiente de la escala en la que se encuentren los datos, siendo solamente útil para comprobar la precisión de predicción entre modelos en una misma serie. En el análisis de regresión, la fórmula de error cuadrado medio generalmente se ajusta para dar una medida de estimación imparcial.

A.6.4 CRITERIO DE R^2

Se denominan también coeficientes de determinación. El criterio está definido por: $R^2 = SCE/SCT = 1 - SCR/SCT$ donde SCE es la Suma de Cuadrados Explicados (calculado con la regresión), SCR es la Suma de Cuadrados de los Residuos y SCT es la Suma de los Cuadrados Total.

La cantidad R^2 se conoce como coeficiente de determinación muestral múltiple y es la medida más común de la bondad del ajuste de una línea de regresión. Igualmente es una estadística descriptiva que calcula la proporción de la varianza de la variable dependiente explicada por las variables regresoras.

BIBLIOGRAFÍA

- ABBASIMEHR, H., M. SHABANI y M. YOUSEFI (2020), «An optimized model using LSTM network for demand forecasting», *Computers & industrial engineering*, **143**, pág. 106 435.
- ACEVEDO, E., A. SERNA y E. SERNA (2017), «Principios y características de las redes neuronales artificiales», *DESARROLLO E INNOVACIÓN EN INGENIERÍA*, pág. 173.
- ARIYO, A. A., A. O. ADEWUMI y C. K. AYO (2014), «Stock price prediction using the ARIMA model», en *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, IEEE, págs. 106–112.
- BALDI, P. y P. J. SADOWSKI (2013), «Understanding dropout», *Advances in neural information processing systems*, **26**, págs. 2814–2822.
- BENALI, L., G. NOTTON, A. FOUILLOY, C. VOYANT y R. DIZENE (2019), «Solar radiation forecasting using artificial neural network and random forest methods: Application to normal beam, horizontal diffuse and global components», *Renewable energy*, **132**, págs. 871–884.
- BOX, G. E., G. M. JENKINS y D. W. BACON (1967), «MODELS FOR FORECASTING SEASONAL AND NON-SEASONAL TIME SERIES.», *Informe técnico*, WISCONSIN UNIV MADISON DEPT OF STATISTICS.
- BREUNIG, M. M., H.-P. KRIEGEL, R. T. NG y J. SANDER (2000), «LOF: identifying density-based local outliers», en *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, págs. 93–104.

- CALHEIROS, R. N., E. MASOUMI, R. RANJAN y R. BUYYA (2014), «Workload prediction using ARIMA model and its impact on cloud applications' QoS», *IEEE Transactions on Cloud Computing*, **3**(4), págs. 449–458.
- CHATFIELD, C. (2000), *Time-series forecasting*, CRC press.
- CHEN, P., T. PEDERSEN, B. BAK-JENSEN y Z. CHEN (2009), «ARIMA-based time series model of stochastic wind power generation», *IEEE transactions on power systems*, **25**(2), págs. 667–676.
- CHEUNG, Y.-W. y K. S. LAI (1995), «Lag order and critical values of the augmented Dickey–Fuller test», *Journal of Business & Economic Statistics*, **13**(3), págs. 277–280.
- CHIA, Y. Y., L. H. LEE, N. SHAFIABADY y D. ISA (2015), «A load predictive energy management system for supercapacitor-battery hybrid energy storage system in solar application using the Support Vector Machine», *Applied Energy*, **137**, págs. 588–602.
- CHO, M., J. HWANG y C. CHEN (1995), «Customer short term load forecasting by using ARIMA transfer function model», en *Proceedings 1995 International Conference on Energy Management and Power Delivery EMPD'95*, tomo 1, IEEE, págs. 317–322.
- CRODA, R. M. C., D. E. G. ROMERO y S.-O. C. MORALES (2019), «Sales Prediction through Neural Networks for a Small Dataset», *IJIMAI*, **5**(4), págs. 35–41.
- DASU, T. y T. JOHNSON (2003), *Exploratory data mining and data cleaning*, tomo 479, John Wiley & Sons.
- DEO, R. C., X. WEN y F. QI (2016), «A wavelet-coupled support vector machine model for forecasting global incident solar radiation using limited meteorological dataset», *Applied Energy*, **168**, págs. 568–593.
- DESPOTOVIC, M., V. NEDIC, D. DESPOTOVIC y S. CVETANOVIC (2016), «Evaluation of empirical models for predicting monthly mean horizontal diffuse solar radiation», *Renewable and Sustainable Energy Reviews*, **56**, págs. 246–260.

- DIAZ, J., J. VUELVAS, F. RUIZ y D. PATIÑO (2019), «Modelo de Predicción de Demanda de Energía Eléctrica Mediante Técnicas Set-Membership», *Revista Iberoamericana de Automática e Informática industrial*.
- ERTUGRUL, Ö. F. (2016), «Forecasting electricity load by a novel recurrent extreme learning machines approach», *International Journal of Electrical Power & Energy Systems*, **78**, págs. 429–435.
- GUERRERO, M., A. LUQUE SENDRA y J. R. LAMA-RUIZ (2019), «Técnicas de predicción mediante minería de datos en la industria alimentaria bajo el paradigma de Industria 4.0», *V jornada de investigación y postgrado de la Escuela Politécnica Superior de Sevilla (2019)*, p 149-157.
- HALABI, L. M., S. MEKHILEF y M. HOSSAIN (2018), «Performance evaluation of hybrid adaptive neuro-fuzzy inference system models for predicting monthly global solar radiation», *Applied energy*, **213**, págs. 247–261.
- HANDFIELD, R. y E. NICHOLS JR (1999), «Introduction to», *Supply Chain Management*, Prentice Hall, Englewood Cliffs, NJ.
- HANNUN, A. Y., P. RAJPURKAR, M. HAGHPANAHI, G. H. TISON, C. BOURN, M. P. TURAKHIA y A. Y. NG (2019), «Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network», *Nature medicine*, **25**(1), pág. 65.
- HO, S. y M. XIE (1998), «The use of ARIMA models for reliability forecasting and analysis», *Computers & industrial engineering*, **35**(1-2), págs. 213–216.
- HOUMLI, R., M. ZMAMI y O. BEN-SALHA (2020), «Short-term electric load forecasting in Tunisia using artificial neural networks», *Energy Systems*, **11**(2), págs. 357–375.
- KATTE, T. (2018), «Recurrent neural network and its various architecture types», *International Journal of Research and Scientific Innovation (IJRSI)*, **5**, págs. 124–129.

- KIM, K.-J. (2003), «Financial time series forecasting using support vector machines», *Neurocomputing*, **55**(1-2), págs. 307–319.
- KIM, M., W. CHOI, Y. JEON y L. LIU (2019), «A Hybrid Neural Network Model for Power Demand Forecasting», *Energies*, **12**(5), pág. 931.
- KONG, W., Z. Y. DONG, Y. JIA, D. J. HILL, Y. XU y Y. ZHANG (2017), «Short-term residential load forecasting based on LSTM recurrent neural network», *IEEE Transactions on Smart Grid*, **10**(1), págs. 841–851.
- LEE, C.-M. y C.-N. KO (2011), «Short-term load forecasting using lifting scheme and ARIMA models», *Expert Systems with Applications*, **38**(5), págs. 5902–5911.
- LIANG, Y., D. NIU y W.-C. HONG (2019), «Short term load forecasting based on feature extraction and improved general regression neural network model», *Energy*, **166**, págs. 653–663.
- LUO, C., J.-G. LOU, Q. LIN, Q. FU, R. DING, D. ZHANG y Z. WANG (2014), «Correlating events with time series for incident diagnosis», en *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, págs. 1583–1592.
- MAITI, M., Y. VYKLYUK y D. VUKOVIĆ (2020), «Cryptocurrencies chaotic co-movement forecasting with neural networks», *Internet Technology Letters*, **3**(3), pág. e157.
- MBAMALU, G. y M. EL-HAWARY (1993), «Load forecasting via suboptimal seasonal autoregressive models and iteratively reweighted least squares estimation», *IEEE Transactions on Power Systems*, **8**(1), págs. 343–348.
- OLEGARIO, C. C., A. D. CORONEL, B. D. GERARDO y R. P. MEDINA (2019), «Towards An Enhanced Backpropagation Network for Short-Term Load Demand Forecasting», *International Journal of Electrical Engineering and Applied Sciences (IJEEDAS)*, **2**(1), págs. 65–70.

- ONG, C.-S., J.-J. HUANG y G.-H. TZENG (2005), «Model identification of ARIMA family using genetic algorithms», *Applied Mathematics and Computation*, **164**(3), págs. 885–912.
- PANG, X., Y. ZHOU, P. WANG, W. LIN y V. CHANG (2020), «An innovative neural network approach for stock market prediction», *The Journal of Supercomputing*, **76**(3), págs. 2098–2118.
- POSE, M. (2009), «Introducción a las Redes de Neuronas Artificiales», *Departamento de Tecnologías de la Información y las Comunicaciones. Universidad da Coruña*.
- REIKARD, G. (2009), «Predicting solar radiation at high resolutions: A comparison of time series forecasts», *Solar Energy*, **83**(3), págs. 342–349.
- SANTANA, J. C. (2006), «Predicción de series temporales con redes neuronales: una aplicación a la inflación colombiana», *Revista Colombiana de Estadística*, **29**(1), págs. 77–92.
- SEN, R., H.-F. YU y I. DHILLON (2019), «Think Globally, Act Locally: A Deep Neural Network Approach to High-Dimensional Time Series Forecasting», *arXiv preprint arXiv:1905.03806*.
- SHANMUGANATHAN, S. (2016), «Artificial neural network modelling: An introduction», en *Artificial neural network modelling*, Springer, págs. 1–14.
- SHERSTINSKY, A. (2020), «Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network», *Physica D: Nonlinear Phenomena*, **404**, pág. 132 306.
- SHU, Y., M. YU, O. YANG, J. LIU y H. FENG (2005), «Wireless traffic modeling and prediction using seasonal ARIMA models», *IEICE transactions on communications*, **88**(10), págs. 3992–3999.
- TAYLOR, J. W. (2003), «Short-term electricity demand forecasting using double seasonal exponential smoothing», *Journal of the Operational Research Society*, **54**(8), págs. 799–805.

- TENG, M. (2010), «Anomaly detection on time series», en *2010 IEEE International Conference on Progress in Informatics and Computing*, tomo 1, IEEE, págs. 603–608.
- VERARDI, V. y C. VERMANDELE (2018), «Univariate and multivariate outlier identification for skewed or heavy-tailed distributions», *The Stata Journal*, **18**(3), págs. 517–532.
- VOYANT, C., G. NOTTON, S. KALOGIROU, M.-L. NIVET, C. PAOLI, F. MOTTE y A. FOUILLOY (2017), «Machine learning methods for solar radiation forecasting: A review», *Renewable Energy*, **105**, págs. 569–582.
- WANG, Y., J. WANG, G. ZHAO y Y. DONG (2012), «Application of residual modification approach in seasonal ARIMA for electricity demand forecasting: A case study of China», *Energy Policy*, **48**, págs. 284–294.
- XU, X., D. CAO, Y. ZHOU y J. GAO (2020), «Application of neural network algorithm in fault diagnosis of mechanical intelligence», *Mechanical Systems and Signal Processing*, pág. 106625.